Journal of Research and Practice in Information Technology

Volume
34
number 1 2002

JOURNAL OF RESEARCH AND PRACTICE IN INFORMATION TECHNOLOGY

The *Journal of Research and Practice in Information Technology* is a quarterly publication emphasising activities that have successfully connected fundamental and applied research in Information Technology with practical application. The journal publishes papers relating to both emerging research and to professional practice and thus contains articles that are of interest both to practicing information technology professionals and to university and industry researchers.

Print Post Approved PP255003/01660

Welcome to the first edition of the JRPIT for 2002. This edition was to have been a special issue on Web-Based Systems. However, as sometimes happens in research, the results are not quite what we expected! Having called for submissions and received and had reviewed quite a number, in the opinion of the editors, the papers which were eventually selected were not closely enough related. We have therefore decided to publish them without the designation of being a special issue.

This is not, of course, to take anything from the papers involved. This edition contains four papers, the first three of which were submitted as a result of the call for papers for the special issue, the fourth as a regular paper.

The first paper by Anita Ferreira-Cabrera and John Atkinson-Abutridy discusses document filtering on the web. They take a linguistic approach, setting up a dialogue which takes into account the context of the user. José Martinez then discusses a formal framework for the browsing of hypermedia databases and, following this, Changjie Wang, Fangguo Zhang and Yumin Wang propose a new system for protecting and anonymising transactions over the Internet. A diverse collection of topics, indeed. The fourth paper from Caroline Allinson discusses audit trails in information systems and reports on an Australian government survey with some worrying conclusions.

John Roddick
Flinders University
and
Michael Schrefl
University of Linz

A Model for Generating Explanatory Web-based Natural-Language Dialogue Interactions for Document Filtering*

Anita Ferreira-Cabrera †
Departamento de Lingüística/Español
Universidad de Concepción, Chile
aferreir@udec.cl

John A. Atkinson-Abutridy
Departamento de Ingeniería Informática
Universidad de Concepción, Chile
atkinson@inf.udec.cl

A computational linguistics approach for Web-based cooperative dialogues aimed at improving results in achieving a successful filtered bibliographic search on the Web is presented. The model focuses on the user's requests by automatically generating language-driven interactions which take into account the context, user's feedback and the initial search results. The main results of a preliminary working prototype to decrease both the number of conversational turns and the amount of information obtained are described.

Keywords: Information Overloading, Dialog Interactions, Natural Language Processing, Document Filtering.

1. MOTIVATION

The increasing use of Web resources in the last few years has created a need for more efficient and useful search methods. Unfortunately, the current mechanisms to assist the search process and retrieval are quite limited, mainly due to the lack of access to the document's semantics and the underlying difficulties in providing more suitable search patterns.

Although keyword-based information retrieval systems has provided a fair first step for the overall process so far, one of the next challenges will be performing those kinds of task in a more precise and smarter manner in order to make good use of user's knowledge (i.e. intentions, goals, etc) to improve the searching capabilities with a minimum of interactions/exchanges (currently, there is no such interaction, it is up to the users to check whether the huge amount of information delivered is useful or not). Our proposal's main claims relies on the fact that those limitations can be partially overcome by dealing with the following working hypotheses:

* This research is sponsored by the Dirección de Investigación, Universidad de Concepción under grant number 98.062.032-1.1D "Processing Interactive Dialog Discourses on a Search Agent Context"

† At the present, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, UK. anitaf@dai.ed.ac.uk

Manuscript received: April 2001
Associate Editors: Michael Schrefl and John Roddick

- To decrease information overload in searching for information implies “filtering” it in an intelligent way in terms of the context and the user’s feedback.
- To make good use of the user’s feedback/knowledge can be useful in obtaining more precise information to be delivered.
- To take into account the linguistic component (and underlying knowledge) as a main working support can assist us in specifying and restricting the real user’s requirements and so capturing user knowledge which is unlikely to be obtained by other means.

Accordingly, our underlying research goal is to generate Spanish interactive dialogues for bibliographical retrieval on the Web based on Natural Language technology and so assist the searching/filtering process with minimum user interaction. To this end, the main working focus will be on enhancing the whole information searching paradigm with both a computational linguistics model and a more suitable search agent to filter and so decrease the information overload. Thus, our approach’s backbone will consist of task-dependent discourse and dialogue analysis capabilities as a part of a major interactive searching system which the user interacts with. While the original approach and implementation was carried out to deal with Spanish dialogues, we provide a model which can be relatively easily adapted to other languages as long as the right grammar and pragmatic constraints are developed (the general task, stages, and high level goals remain).

Our experiments, conducted in the context of a filtering system for Web documents, demonstrate the promise of combining *Natural Language Processing* (NLP) techniques with simple inference methods to address an information searching problem. In what follows, we first motivate our work and discuss previous related work. Next, we discuss the distinguishing features of our approach along with the analysis methods and representation used. Finally, details of some experiments are described.

2. RELATED WORK

Nowadays, many search engines use automated software which searches the web and obtains the contents of each server it encounters, indexing documents as it finds them. This approach results in the kind of databases maintained and indexed by services such as *Alta Vista* and *Excite*. However, the problems which users can face when using such databases are beginning to be well documented:

- Relevant Information: from the information obtained (i.e. references to documents), users can spend a long time trying to check whether those results contain what they have been looking for.
- Information Overload: the amount of information and wide coverage is often so huge that most of it is usually abandoned and just the first supposed-to-be important references are kept.
- Representation: the information search/retrieval is based on texts/documents as a bag of keywords and all the further similar calculations rely on them.
- User’s Feedback: there is no interaction with the user in the searching process in order to check whether her/his requirements have been fulfilled or not. So, it is assumed that all the information is useful, but it is unclear how.

Intelligent searching agents have been developed in order to provide a solution (mostly) to these problems (Levy and Weld, 2000). These agents can use the spider technology used by traditional web search engines, and employ this in new kinds of ways. Usually, these tools are “robots” which can be trained by the user to search the web for specific types of information resources. The agent can be personalised by his/her owner so that it can build up a picture of individual profiles or precise information needs. An intelligent agent can also be autonomous – so that it is capable of making judgements about the likely relevance of the material. Another important feature is that their usefulness as searching tools should increase the more frequently they are used. It will learn from past experiences, as a user will have the option of reviewing search results and rejecting any information sources which are neither relevant nor useful. This information will be stored in a user profile which the agent uses when performing a search. Thus, an agent can also learn from its initial forays into the web, and return with a more tightly defined searching agenda if requested. Representative current tools using this technology include: *FireFly* (music and film recommendation system which uses intelligent agents to build up profiles of user preferences using collaborative filtering), *Webwatcher* (learns from the user's reactions), *Letizi* (tries to anticipate the user's preferences based on its experience and previous behaviour) and so on. Nonetheless, using any current approach raises some practical problems for end-users:

- To navigate through the Web is a time-consuming and sometimes troublesome task for users, especially when there is a huge overload of the information delivered.
- To specify what is to be searched for is not an easy task, users do not know what they really want and therefore, there is no efficient way to assist them in that endeavour.
- To get better results to provide the system with different choices, would cause a huge waste of time and effort.

A common constraint of many search systems is the lack of a deeper linguistic analysis of the user's requirements and context to assist him/her in getting a more specific view about what he/she really wants and so the quality of the information being obtained. There are a few cases which, in a very basic way, deal with these limitations, such as *AltaVista*, which has considered both statistical variables and linguistics issues (i.e. lexicalisation) in the search process, however, the approach is still preliminary and mainly based on the documents as a bag of keywords (Berry and Brown, 1999) – an underlying assumption in many state-of-the-art Information Retrieval (IR) systems, indeed. Unlike IR, information filtering (IF) has recently become popular. Depending on how the user selects the documents, three kinds of IF systems can be distinguished:

- *Cognitive*: select documents based on their contents' features.
- *Social*: select documents based on other users' recommendations.
- *Economical*: select documents based on some cost/benefit metrics to the user.

Several approaches have been used to get the document's "semantics" (in a restricted way, though), including *Oval* which uses a keyword-based strategy, *Foltz* which uses LSA (Landauer, *et al*, 1998) to filter news articles, *INFOSCOPE* which uses rule-based agents to watch the user's behaviour and then to make suggestions, *MAXIMS*, for collaborative electronic filtering (Maes, 1993; Maes, 1994), *WebWatcher*, a Web filtering system which learns user's preferences and highlights interesting links on web pages that it visits, and so on. Both cognitive and social approaches are suitable ones to extract documents. The difference relies on the fact that, depending on the application domain, one is more advantageous than the other. Whereas social filtering is more appropriate when the information obtained is used to keep it updated in some environment, a cognitive approach is better when the information gathered is based on some specific topic, regardless of who the other users are. Therefore, learning and adaptation capabilities get more importance in an IF context rather than IR because of the underlying environment's features: IF systems are used by a huge group of persons, most of whom are not motivated information searchers, and so their interests are often weakly defined and understated (Ram, 1991). For this, IF systems should be able to answer the user's dynamic interests in a more intelligent way (Levy and Weld, 2000).

On the language side, part of these problems could be overcome either by extracting "more" knowledge from what the users are looking for or by interactively generating more explanatory requests to have users more focused in their interests. Although some research has been carried out using NLP technology to capture user's profiles, it has only been in very restricted domains which use *WordNet* or more primitive resources, and have, as main focuses, the information retrieval problems and issues concerning concept generalisation to be able to proactively act on users' requirements (Bloedorn and Mani, 1998). Deeper approaches to this kind of interaction could be used by making good use of NLP. In particular, Natural Language Generation (NLG) techniques can be used to allow the system to produce good and useful "dialogue" with the user. Thus, an important issue at this stage will be in decreasing the number of generated conversation/interaction turns in order for the user to obtain the information (i.e. references to documents) he/she is looking for.

Other research considers this kind of dialogue discourse (user-system conversation turns) as a whole structured into different levels (Moore, 1995; Reiter and Dale, 2000) in which communication constitutes an indirect action to accomplish the goals, and the pragmatic plays an important role as long as they can be established in both form and ground.

Research on NLG has strongly evolved due to the results obtained in the first investigations. Since then, the task of establishing and processing the discourse content has been privileged (Dale, *et al*, 1998; Reiter and Dale, 2000), so an important issue here concerns the discourse planning (Cohen and Perrault,1979) in which, based on the speech acts theory (Austin, 1963; Searle, 1975) that linguistic concept is incorporated into the description of computer systems producing plans which contain sequences of speech acts (Cohen and Levesque, 1990). When discourse involves dialogue situations, NLG systems are able to return the conversational turn, to reply according to the subject's knowledge degree participating in the discourse, to react from mistakes and to deal with unexpected reactions from the hearer (Moore, 1995; Chu-Carroll, 2000; Rich and Snider, 1998; Jurafsky and Martin, 2000) and so on. The tasks of NLG are now commonly seen to involve (Dale, *et al*, 1998):

- *Content Determination:* deciding what to say which impacts at both macro (how the content of a sentential text or of a turn in a dialogue, can be determined), and micro (how the content of appropriate referring expressions can be worked out) levels.
- *Text Structure:* concerning with elucidating mechanisms to determine the most appropriate structures to use in particular circumstances.
- *Surface Realisation:* in which the interest is that, once the content of the sentences has been determined, it has to be mapped into morphological and grammatically well formed words and sentences.

In order to model NLG based dialogue interactions, some approaches have been identified (Wright, *et al*, 1999; Lochbaum, *et al*, 1990), including:

1. **Dialogue Grammars:** Phrase structure grammar based finite state machines in which dialogues are considered to be a sequence of adjacency pair-like speech acts.
2. **Dialogue Plan Based Theories:** based on the underlying assumption that utterances are not only word sequences but also speech acts update actions, such as illocutionary acts to request, inform, suggest and so on. It also assumes human beings are not performing those acts in

random ways, but planning their actions aimed to achieve some goal. In terms of communicating acts, those actions imply mental changes on hearers which have to find out and to appropriately answer the speaker's plan.

3. **Dialogue as Cooperation:** which is an activity carried out by sharing agents (Rich and Sidner, 1998) rather a product of the interaction between plan generators. From this, participants are required, at least, to have a commitment for understanding each other.

3. OUR APPROACH TO SEARCH-DRIVEN DISCOURSE PROCESSING

In order to deal with the difficulty of coming up with suitable profiles or likes, the current filtering systems allow the users to specify one or more sample documents as reflective of his/her interests (Tong, *et al*, 2001; Ram, 1991), instead of requiring direct explicit definition of interest, whereas others attempt to learn that from the user's observed behaviour. However, that approach is impractical when users are not focused in what they really want, when they do not have document samples matching their requirements, and even when they go through the Web without a clear guiding point which can lead the underlying agent to wrong directions and inferences.

Instead of providing samples or going through the Web looking for the information, we propose a new approach in which search requirements are focused by using a dialogue-based discourse interaction system with the user in order to capture his/her specific interests, and the search itself in a way that there is knowledge enough to filter the initial search results.

The approach we have developed is designed to meet a variety of requirements. It must be scalable to large collections of documents. A system using this approach should be able to quickly determine the user's needs through a combination of user provided information and feedback, all of these based on natural language interaction.

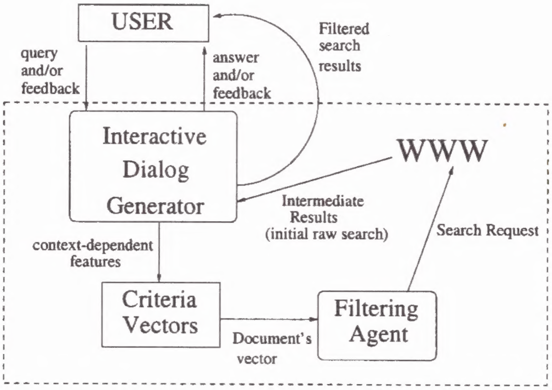


Figure 1: The Overall Search-driven Dialogue System

Our proposal for filtering through discourse processing is shown in Figure 1. The operation starts from Natural Language (NL) queries provided by the user (i.e. general queries, general responses, feedback, confirmation, etc) and then passed to the discourse/dialogue processing phase which will generate the corresponding interaction exchanges (turns) in terms of NL utterances, to arrive into a more elaborated and specific search request. As the dialogue goes on, the system generates a more refined query which finally is passed through a search agent. The search's results are delivered to the user as long as they have been appropriately filtered, which depends on the previous interactions, the user's context and the features extracted from the queries.

The next sections describe how the main stages have been conceived.

3.1 Experimental Methodology

Dialogue models can be built from data obtained using the *Wizard of Oz* (WOZ) technique. In our proposal, we have used that method in order to develop and to test the dialogue models.

During each interaction, the human (*the Wizard*) simulates a system which interacts with the users to whom we make believe they are interacting with a system which handles natural language in a real way. Then, the dialogues are recorded, annotated and analysed with the ultimate goal of improving the dialogue model and therefore, the interaction. The interactive process continues until the model meets either the expectations or the design requirements.

In the actual experiments, WOZ has been used to get a significant dialogue corpus which allows us to analyse the transcriptions and to establish a dialogue structure based model that will support the planning and generation of interactive explanatory and descriptive discourse.

By using Java, a WOZ simulator called "ENCUENTRA" ("find") which incorporated search facilities, was implemented. Then, a group of subjects was told to search for information on the Web using that "tool". Furthermore, the interactions between users and system (actually, the hidden expert who interacts with the users and search for the real information) were automatically recorded and used for the linguistic analysis. Thus, the communicating situation using the simulator became a three-phase activity:

1. Users engage in a dialogue with the computer in order to make their search query more precise.
2. The dialogue is monitored by the expert, and as the dialogue proceeds, he builds the answers.
3. The answer is sent to the user.

On the other hand, users have three access areas available to interact: the writing area, the reading area (either from the results or from the WOZ answers), and an area to receive and to explore the search's results.

Based on the information available, user and "system" are able to continue the refining process until the user's need is met or he retracts from searching any longer.

In practice, a threshold of 20 minutes was considered to check for the user's communicating goal accomplishment with a total number of 20 non-expert subjects being involved (i.e. real samples). Next, the sample was divided into four groups, in which the first three groups were randomly selected whereas the fourth one was constituted by graduate students of linguistics. Furthermore, they had to perform the search and then to provide definitions for "explanation" and "description" (as the final model should be able to generate both descriptive and explanatory discourse depending on the context, situation, and search's results) discourse in terms of the results obtained. Specifically, the groups were told to perform the following tasks:

- The first group was required to "talk about" the search's results: *Could you talk about your search's results?*.
- The second group was required to "describe" the search's results: *Could you describe your search's results?*.
- The third group was required to "explain" the search's results: *Could you explain your search's results?*.
- And, the fourth group was required to do the three above tasks and to answer some questions stated by the system, including:

- *What do you understand by explaining or describing? Give me an example.*
- *What do you understand by describing the search's results?*
- *What do you understand by explaining the search's results?*
- *What is the difference between describing and explaining the results?*

At the end of those tasks, and in order to validate whether the subjects were able to establish differences in producing their discourses (i.e. explaining, describing, etc) based on the current communicating situation, some advanced students of linguistics were asked to perform the different tasks but in the same session.

In addition to the transcriptions and the dialogue structures produced, several parameters which guide both explanatory and descriptive discourse about the search were produced (further details in Ferreira, 1998), including: documents' date, language, source place, kind of WWW page (ie. research, business, etc) which were also produced in developing the NL generation module.

3.2 Interactive Dialogue/Discourse Generator

The discourse generator relies on a number of stages which state the context, the participants' knowledge (user and system) and the situation where the dialogue discourse analysed by the system is being conceived (i.e. interaction aimed to search for information on the Web). It also considers a set of modules in which input and output is delimited according to different stages of linguistic and non-linguistic information processing defined by the dialogue. This phase is strongly based on the linguistic proposal of a model for discourse processing by Van Dijk (1995) and the discourse approach by Schiffrin (1987) regarding the components of interaction and action.

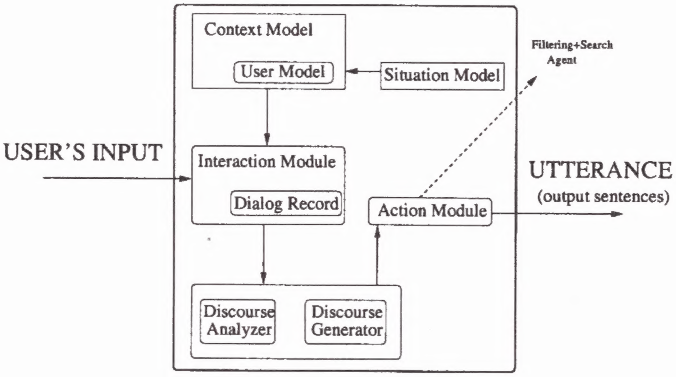


Figure 2: The Interactive Dialogue/Discourse Processing Stage

In Figure 2, the proposed model design to generate discourse on bibliographic search on the Web is shown (Ferreira, 1998). It starts with the user's input (NL query) and produces either an output consisting of a NL conversation exchange to guide the dialogue and to have the user more focused (if the dialogue is in a intermediate interaction) or a search request to be passed to the search agent (if the dialogue is in a final stage and all the information has been filtered).

In order to better understand the approach, the underlying working has been separated into different components/modules as stated in Figure 2 and described as follows:

- **The Context Model** deals with the information regarding the dialogue's participants. This is, the "user" who needs information from the Web and the "system" which performs the search.

This model states the kind of social situation called "bibliographical queries on the Web" and the participants' goals: "find out information about some topic" (user's) and "assist the user on achieving her/his goal through searching and collaborative dialogue" (system's). Here, the **User Model** includes knowledge about the user (i.e. features) with whom the system will interact in a collaborative way. The user model's outcome will become the type of query which the system expects as input, this is, a kind of bibliographical "query" on the Web. The information regarding the communicative situation's characteristics in which the dialogue discourse is embedded, is established on the **Situational Model**. Because of the communicating interaction's purpose, conversations must be limited to the requirements and constraints requested by such situation. This implies using both some status records and utterance structures (lexical, syntactic, semantics, pragmatics) to represent it.

- **The Interaction Module** is based on Grice's cooperative principle and collaborative maxims (Grice, 1975) and involves two-position exchange structures such as question/answer, greeting/greeting and so on. These exchange structures are subject to constraints on the system's conversation, regarding a two-way ability to transmit through the keyboard, suitable and understandable messages as confirmation acts. Constraints have also to do with the speech acts chosen at some time given on the dialogue process.

  All this information and that related to the different interactions between system and user during the dialogue is stored in a *Dialogue Recording Module* where the goal is to keep the dialogue coherence between the system and the user's input.

- **The Discourse Analyser** receives the user's query and analyses the information contained in order to define the conditions which can address the system's response generation. This module's outcome is the query both recognised and analysed by the system. Furthermore, recognition and interpretation is controlled by two main analysis modules which process the linguistic knowledge and interact with each other: semantics and pragmatics analysers (in addition, there are underlying morphological and syntactical stages but they have been let apart due to length constraint).

  The semantic macro-structures (i.e. propositions), which constitute the **semantic module**, involves selecting an argument from some property or action associated to them. Thus, the process of defining the propositions is guided by the modules of interaction and action, and so its outcome is an appropriate semantic proposition for speech act generated from the pragmatic analysis stage.

  The **Pragmatic Module** aims to establish the kind of speech act suitable to both the dialogue structures component and the constraints and conditions stated. It concerns the information from the *situation model* and the *context model*. In addition, it involves the semantic content suggested by the *semantic module* and the discourse coherence based on the *interaction module*, so its outcome is the previously defined speech act suitable for the current communicating goal.

  Each pragmatic function obtained from the speech acts stated in the search, involves making explicit some specific speech act. On the other side, each semantic proposition states the speech act's semantic content. This content is expressed as a set of semantic functions such as agent, event, object, instrument and so on.

- **The Discourse (dialogue) Generator** involves both the information from the search agent's *information recording module* and that coming from the *dialogue recording module* to produce a coherent utterance on the current dialogue sequence. As a first output, the module generates a question to the user about the information needed to produce the corresponding utterance to the dialogue's conversational turn.

Dialogue starts by generating the kind of utterance "query about information requested by the user" (quite general at the beginning). Next, the system considers two possible generations: a specific query for the communicating situation (`what topic do you want to search for?`) and a general one on the context of the different kinds of information available on the Web (`what kind of information do you need?`). Then, further user's requests can be divided into four general higher groups: request for information, positive/negative confirmation, specification of features, specification of topic, etc.

The discourse analyser processes the user's input and gets the information needed to be provided to the search agent which performs the selected search itself. From the information obtained (i.e. references to documents) the generator will be able to address the dialogue towards a explanatory generation (there is also a kind of descriptive discourse/dialogue but due to length issues, it has not been considered here) according to some basic underlying criteria identified in the preliminary experiments.

Later on, the discourse generator will produce its output according to the results of the search, so a first step is to establish the high level kind of generation to be performed. However, since the generation is fully unstructured and too broad some decisions must be made to better guide the generator from pragmatic up to lexical levels. Accordingly, from experimental studies with the human subjects some additional rules to restrict the production were learned such as: Let R e the number of references obtained by the agent and the following cases are processed: $R > 100$ (too general), $30 < R < 100$ (other issues like "language" – pages written in a different language, type of documents and so on, are taken into account in filtering), $R < 30$ (right enough to show the results to the user, high level pragmatics are invoked). Furthermore, the discourse analyser makes some single decisions according to the number of keywords which express the user's topic of interest and their context, among other issues. Then, this generator can produce two kinds of explanatory utterances: one aimed at having a more detailed specification of the user's query: *Your query is too general, could you be more specific*?, or one which requires the user to state some feature of the topic being consulted: *I found too much information/I found N references to documents about that topic, in which one are you most interested?*. The discourse analyser again performs the analysis on the user's specific input in order for the agent to do the search. In doing this, the information recording module stores the query specification or the thematic topic provided by the user. Then, the agent searches again for the information on the topic requested. Now, the generation leads to descriptive sentences. From the information obtained from the intermediate search results and the user's context (confirmation, negation, request, etc), the discourse generator produces three kinds of descriptive utterances providing different choices to the user, including:

1. To provide the results in a specific language: *"the information is written in different languages, do you prefer them in English?"*,
2. To show all the document references obtained: *"There are twenty items about that topic, do you want to check all of them?*",
3. To show the results according to some frequent parameter: *"I found information about research groups, courses and personal pages, are you interested in some specific issue?"*.

The corresponding search action is performed by the action module (Figure 2) which receives the information analysed from the discourse analyser, so the recording module stores the response provided by the user aimed to do that action and to keep the dialogue coherence.

At this point, the (discourse) analyser processes the user's response in order for the generator to produce an output confirming or expressing the action done (i.e. "Did you find what you were looking for?").

In order to verify whether the communicating goal has been achieved or not, the analyser processes and checks the input. Next, the recording module stores the positive or negative results from the analyser. If a positive response is obtained then the system will generate a sentence to give the user the opportunity to choose another topic to do a new search. Otherwise, the option of searching for another topic related to the recently found one will be produced, starting from the pragmatic level.

It is worthwhile to note that utterances are automatically produced from the groundings, so the pragmatic constraints, high level goals and the previous rules are only part of the strategy to restrict the deep structure of the output, as shown in further examples, there are many ways how an explanatory generation or request for more detailed information can be generated.

The overall process starts by establishing a top goal to build down the full structure in the sentence level. In general, the subsequent (sub) goals have been divided into linguistic functions aimed at initiating the dialogue, answering a query, asking for discourse goal achievement, requesting new topic/subject, etc. Once the goal has been identified, the corresponding speech acts are produced to guide the further generation. For instance, from the starting point, the production and grammar rules would look like:

```
conversation_turn_1 -> Speech_Act_1 ... conversation_turn_2 -> Speech_Act_2
...
Speech_Act_1 -> Interrogation_Act_1
...
Interrogation_Act_1 -> Proposition_question_9
...
Proposition_question_9 -> REQUEST_FOR_OBJECT + EVENT
...
Description_Proposition -> AGENT + EVENT + MEANS
...
AGENT -> NOUN_PHRASE_1
(surface generation starts at this point)
...
MEANS -> NOUN_PHRASE_10 | NOUN_PHRASE_4 ... EVENT -> VERB + VERB_PHRASE
(further linguistic features and constraints, lexicon, etc).
```

Selecting the top-down goal will depend on the context, the participants, and the search results. From the overall strategy, the results can contain utterances/sentence with the same deep structure but different surface realisations (i.e. according to some context, there are different ways to express the same goal).

3.3 Search and Filtering Agent

Unlike traditional search engines or IR systems, we have designed a search agent which does not deliver all the information to the user from the first time but also that information is used to feed both the system's knowledge and the user's request and queries. As the interaction goes, the agent refines the request and filters the initial information already obtained until a proper amount of information can be displayed at the end of the dialogue.

This **Filtering Agent** as stated in Figure 1, is a compound of three components: the information searcher itself, the criteria analyser which processes the information found according to some parameters (criteria), and the information status record which keeps the information about the results of the analysis to be accessed by the discourse generator later to produce the output sentence which fits the current conversation dialogue sequence and search constraints. By "criteria", it is meant the kind of underlying representation stated for the documents and the user's profile which is similar to that used in IR but it has been enhanced with vector based specific purpose features to support the expressiveness needed. Both documents (i.e. web pages) and user's queries (or any other dialogue's information depending on the discourse exchange at that time) are represented in a multidimensional space so when a query is processed it is then translated into a pattern representing a criteria vector. Then, by using distance metrics and some existing engines, the appropriate documents are retrieved. Documents so structured are represented as: $V_i = X_0\ X_1\ ...\ X_n$ where X_i states the value extracted from the users or the search results for the *i-th* criterion/field of the document/vectors being obtained.

Those criteria represent important context information related to Web pages found and so they can be useful in training the patterns and filtering the results. Initially, criterion X_0 will concern the subject or input's topic and the rest of the vector will remain empty (as the dialogue proceeds and new search results are obtained, these slot are gradually being filled).

From these criteria, the dialogue samples and the context information, it was possible to extract and synthesise the most frequent patterns used to decide whether some information is useful or not. Some of them included (not necessary in order of importance): *URL Address of the Web page being selected, Documents' author, Language which the document is written in, Document's source country, Type of document/page (commercial, education, etc), References/documents related to events, Technical documentation, Research groups, Products and Services*, and so on.

For example, component X_4 could store the criterion "language", so if a further interaction would state that the current search concerning documents in *Spanish*, the slot X_4 would be filled with the value *Spanish*, and that would take part of a further search for some documents.

Every time the system produces a "question/feedback" to the user, it becomes a known criterion so more features will be filled until there is knowledge enough and is suitable to perform a more specific search. In addition, each criterion has some "weight" which represents its contribution to a defined document or the importance of some features over others.

Whether the criteria are filled with information from the dialogue or from the current intermediate search, the agent (previously trained) takes those vectors and then makes the search request from the Web. If no further filtering can be done, the results are shown, otherwise, new requests from the dialogue context are issued to the agent. In addition to the pragmatic and discourse issues related to the dialogue, decisions on specific actions to be taken given situation context knowledge depend on two kinds of ground conditions: *Information on the documents' slots/criteria (if any),* and *Simple inferences drawn when the previous information is not enough or it is not available*. The later has to do with a rough statistical confidence of performing certain actions given some situation (i.e. criteria value, missing values, etc). This inference's result has two basic consequences: one affecting the information filtered and the other assisting the sentence generation to look for criteria/features missed or incomplete. The general basic algorithm to calculate confidence levels and then to select actions to be taken is stated below:

```
ALGORITHM: Obtain Action for criteria
Input : document_vector, situation
Output: type of Action

BEGIN
   CF <-- Probability of action given some situation
   IF (CF >= 0.5) y (CF<=0.96) THEN
      Agent make suggestion with confidence CF
      IF (positive_feedback_from_user) THEN
         Update occurrences (action,situation)
         Add new situation to the list of candidates situations
         Search and Filter documents according to situation
         (new criterion's value found)
      END
ELSE
   IF (CF >0.96) THEN
      Agent make question according to context information
         and action given by higher situation
      Update frequencies of action given situation
   ELSE
      IF (CF=0) THEN
         Action <-- Go on with Dialogue
      ELSE
         Action <-- request for further information
                    (Higher pragmatic level invoked)
END
```

In practice, those actions are translated into high level goals of pragmatic constraints which cause a particular kind of dialogue to be generated (i.e. question, request, feedback,..).

4. RESULTS

The results of the system can be stated in terms of two main issues regarding our initial goals and hypotheses: one regarding the kind of utterance automatically generated by the system which proves the search driven discourse generation to be feasible. A second issue concerns the benefits of using this kind of interaction to decrease information overload and therefore, the time spent by the user looking for information.

On the dialogue processing side, a prototype system was built in which the discourse generator was implemented in the **SnePS** framework (Shapiro, 1995) which is based on extended *ATN* dealing with semantic nets and high level construction linguistic rules, and a restricted medium-size NL interface for user's input parsing was designed using the GILENA NL interfaces generator (Atkinson, 1998) which allowed us to link the application with Web resources.

In processing the rules implemented in the discourse generator, several discourse inputs were used, so generating each rule involved producing the corresponding utterance. The analysis of results was based on the generation of 1000 dialogue structure samples obtained from the discourse processing task carried out by the system. The discourse manager was able to produce dialogue structures and to interact with the user starting from communicating goals (in terms of actions) as follows (**S** stands for the system's output, and **U** for the user's input, with the corresponding English translations):

ACTION: the system generates a kind of sentence/query on a topic requested by the user:

```
S: sobre que topico deseas revisar?/what are you interested about?
U: sobre linguistica/about linguistics
```

ACTION: "Specification of the results of the search topic" (the generation will be on a conversational turn of request for more specific information about the topic):

```
S: tu consulta es muy amplia puedes delimitar la mas por favor/ your query is
   too broad, could you please be more specific?
U: bueno/Ok
```

ACTION: Search's results and dialogue context dependent generation (the kind of utterance generated is fully adaptive so it varies from one interaction to the other):

```
U: La informacion esta en distintas lenguas, la prefieres en Espanol? /
   the information obtained is in different language, do you prefer it in
   Spanish?
..
U: Hay veinte referencias sobre este tema las quieres revisar todas?/
   There are twenty references about that topic, do you want to check all of them?
   ..
U: Halle informacion sobre grupos de investigacion, cursos, etc que aspecto te
   interesa?/
 I found informartion about research groups, courses, etc, what are you interested
   in?
```

ACTION: The generation of the sentence/query on achieving the goal is restricted:

```
S: averiguaste lo que necesitabas?/
   Did you find what you were looking for?
U: no
```

ACTION: "Specification of the discourse goal achievement" (a kind of question requesting other information related to the topic is generated):

```
S: necesitas mas informacion sobre este tema?/
   do you need further information about that topic?
U: bueno/yes
..
S: Halle bastante informacion acerca de este tema, que rasgo mas preciso
```

prefieres revisar?/
I found too much information about it, in which feature are you most interested?
...

ACTION: "Dialogue conclusion" (two kind of utterances are generated: a question-utterance to achieve the discourse goal and the ending utterance to conclude the dialogue):

S: encontrastes lo que buscabas?/
did you find it?
U: no realmente / not really
S: lo siento para otra vez sera ... hasta pronto /
I'm sorry, next time. See you soon

On the filtering and interaction side, the system performance was analysed regarding the experiments evaluating the number of conversational turns in the dialogue necessary to get a more precise requirement and filtered information against the number of references/documents which matched those requirements. Initially, the universe of possible candidates became more than 30.000 document references but due to this study's purpose, the scope has been then reduced to not more than 1000 references.

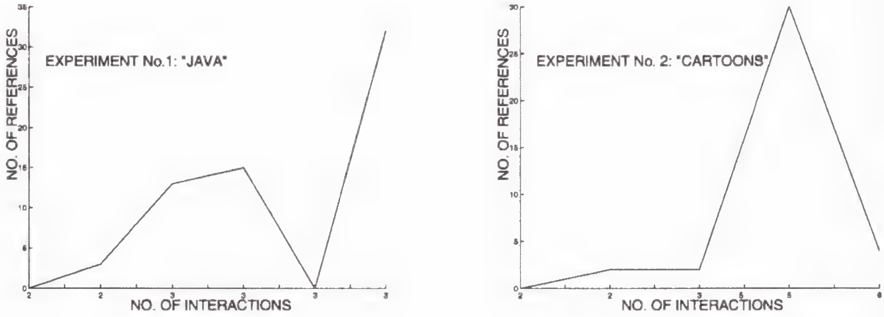


Figure 3: Interactive Experiments

Two experiments were carried out (Figure 3): In the first one, one of the main interest topics concerned the focus **Java** (not the keyword), and the second, **Cartoons**. To better understand the analysis, each interaction is defined by one or more dialogues (exchanges) between user and system.

Interactions in experiment No. 1 showed an increase in the number of documents matched when more than three turns are exchanged. It does not come up by chance or numerical trends: for the same number of interactions (i.e. five), different results are shown mainly due to the adaptive way how the dialogue continues. This is, the context and kind of questions by the agent are changing depending on (among other issues) the situation and the document's contents. Different results were obtained for the same number of interactions because the kind of document searched for was changed as other features were restricted. A similar situation occurs when the dialogue states a constraint regarding the language, in which case, most of the references matched (or even no references) or were produced at all.

In the second experiment, something similar happened. Even in dialogues with three exchanges, sudden increments were observed, going up from 1 to nearly 35 resulting references. That change is due to an inference drawn by the agent and a user's restriction related to the document's nature he/she is looking for.

From both experiments, it can be seen that there is an important drop in the results obtained with a minimum of conversation turns due to constraints on the nature of the information finally delivered. Our prototype agent took the previous issues into account so there are some classes of high level requests which are more likely than others depending on the current context.

5. CONCLUSIONS

In this work, an approach and cooperative strategies to deal with the problem of information overloading when interactions with the user are taken into account has been presented. Although the original design was intended to handle Spanish NL interactions, it should not be too difficult to adapt it to others languages and needs.

Initial hypotheses regarding user's feedback and the search agent's inference capabilities have been experimentally tested in medium-size situations. The analysis could have gone deeper from an IR point of view, however our goal was to provide an integrated and more global view in order to put together all the referred elements rather than concentrating on typical IR metrics as they mainly involve the surface side of the searching/filtering process (feedback loop is never considered).

From the underlying experiments, we claim that a lot of time can be saved if we are provided with the weighted features usually presented on the information retrieved depending on its importance degree or usage. In any case, interactions (form and content) will strongly rely on those factors, however, it should not leave user's contributions apart from the decisions being made (and later on, generated) by the system.

In addition and despite the moderated complexity of the experiments and design constraints, the issues which have been identified should not drastically change through more advanced requirements and implementations (i.e. different language, different search capabilities, etc).

From a language-centered viewpoint, the current model based on dialogue interactions has proved to be a novel and interesting work methodology to deal with more specific information searching requirements in which both designing and implementing a NLG system can easily be adapted to the current communicating situation. Even though there is a lot of NLG systems, as far as we are concerned, this is the first attempt to integrate these technologies to address the problems of searching and filtering on the Web.

Compared to similar approaches using NLP such as PIES (Ram, 1992), the following features can be highlighted in our model:

- The user's interests and goals are obtained as the dialogue proceeds whereas in PIES that kind of information (i.e. interestingness and relevance) is established in advance by providing different parameter values.
- The underlying working approach in PIES is on analysing the user's profile to prune part of some "story" in order to reach a final answer. Since our "stories" are real documents extracted from the Web, we cannot afford understanding the complete set of texts. Instead, our model deals with the dialogue as a way to extract important knowledge from the user and then to filter the proper references.

Over the last few years, a different approach to incorporate NLP technology on Web search and retrieval has emerged, which has been based mainly on the TREC track on CLARITECH Natural-

Language (Voorhees, 1999), which recently has turned into Question-Answering (QA) systems. However, they address the problem of using linguistic processing into different levels, and using additional resources (i.e. WordNet, pos-tagging, sense resolution, etc) to assist the document retrieval and indexing. There is no dialogue at all and the effort is centered in getting precise documents rather than capturing the user's needs. According to their studies and experiments, it is claimed there are some problems to be dealt with, which fit fair enough our model's main contributions:

- *Linguistic Techniques must be essentially perfect to help:* it has been told that small imprecisions in tasks ranging from word sense disambiguation to co-reference resolution, could produce a disaster in terms of degrading the IR system effectiveness. Indeed, if the focus is on the document itself, that kind of assertion can become true. However, we propose an approach in which important knowledge from the interactions with the user is captured and then the search itself is carried out. Even if we are provided with very sophisticated NLP methods to extract the right information, it is not very useful as we do not know in advance what the user is looking for or why he/she is doing that.
- *Queries are difficult:* of course, the scope and complexity of the processing needed to deal with the user's queries is far from being simple. Nevertheless, it has been practically proved that managing dialogues and queries in restricted domains and tasks can substantially handle that difficulty. Indeed, our "restriction" in terms of complexity, relies on obtaining real dialogue samples of subjects interacting in real situations for achieving a specific communicating goal.
- *Nonlinguistic techniques implicitly exploit linguistic knowledge:* It is claimed that even if the NLP task is perfect, linguistic techniques may provide little benefit over proper statistical techniques. From our point of view, that is not completely true, because there is a lot of knowledge we are exploiting which has to do with the context, interests and goals, and the user's behaviour along the dialogue. As seen in the results, we have been able to filter information which no additional IR facilities could do, so the differences lie in the adaptive way how the dialogue goes on.

On the other side, there are some important similarities with the current QA technology (Harabagiu, *et al*, 2000; Voorhees, 2000), but our approach exhibits important additions as well:

- While the main purpose of QA systems is to extract the correct answer for a question (actually, the document extracted from the Web which contains the answer), they need to be provided with a lot of sample information in advance: they know the kind of questions the user can make, there is a predefined set of target documents where the answers are supposed to lie in, there are a few constraints which restrict the scope of the likely questions, and so on. For this, the filtering process should be perfect as long as they know everything possible to get the answer. In our approach we are not restricted to some particular sort of question or domain and, therefore, we do not even know whether we will be able to find the information until the dialogue is over, which makes the process more flexible and well suitable to different domains (with the same "search" task in mind).
- Since most of the QA task is fixed (and the participants have time enough to make changes to their systems), there is only one possible answer (or a few of them) for every question and there is no feedback possible. This is, one question implies one likely answer, so one cannot go back to refine the request or the results: if the answer is wrong then the participant is penalised. Accordingly, the scenarios so far are still artificial. Instead, we provide strategies which allow us to carry out a dialogue and to refine the answers according the real search results.

- Finally, our model can be considered an addition and improvement in both representation and processing in order to fill in the gap between two different extremes: traditional IR systems and QA systems, in terms of representation, task-oriented dialogue, general-purpose questions, and filtering using linguistic knowledge and context information.

REFERENCES

ATKINSON, J. (1998): The design and implementation of the gilena natural language interfaces specification language. *ACM SIGPLAN Notices*, 33(9):108-117.

AUSTIN, J. (1963): *Performatif Constatif, Philosophy and Ordinary Language*. Oxford University Press, England.

BERRY, M. and BROWNE, M. (1999): *Understanding Search Engines: Mathematical Modeling and Text Retrieval.* SIAM.

BLOEDORN, E. and MANI, I. (1998): Using NLP for machine learning of user profiles. *Intelligent Data Analysis.*

CHU-CARROLL, J. (2000): Mimic: an adaptive mixed initiative spoken dialogue system for information queries. *Proceedings of the 6th Conference on Applied Natural Language Processing*, Seattle-USA.

COHEN, P. and LEVESQUE, H. (1990): Performatives in a rationally based speech act theory. Technical Note 486, SRI International.

COHEN, P. and PERRAULT, C. (1979): Elements of a plan-based theory of speech acts. *Cognitive Science*, 20(3):177-212.

DALE, R., DI EUGENIO, B., and SCOTT, D. (1998): Introduction to the special issue on natural language generation. *Computational Linguistics*, 24(2).

FERREIRA, A. (1998): *Generating Descriptive and Explanatory Discourse based on a Computational Linguistics Model (in Spanish).* PhD thesis, Catholic University of Valparaiso, Chile.

GRICE, H. (1975): Logic and conversation. *In Syntax and Semantics*. Cole Morgan.

HARABAGIU, S., PASCA, M., and MAIORAMO, S. (2000): Experiments with open-domain textual question answering. *Proceedings of COLING-2000*, 292-298.

JURAFSKY, D. and MARTIN, J. (2000): *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

LANDAUER, T., FOLTZ, P., and LAHAM, D. (1998): An introduction to latent semantic analysis. *Discourse Processes*, 10(25):259-284.

LEVY, A. and WELD, D. (2000): Intelligent internet systems. *Artificial Intelligence*, 11(8):1-14.

LOCHBAUM, K., GROSZ, B., and SIDNER, C. (1990): Models of plans to support communication: An initial report. *Proceedings of AAAI-90, Boston.*

MAES, P. (1993): Evolving agents for personalized information filtering. *Proceedings of the Ninth Conference on Artificial Intelligence for Applications '93, Orlando, Florida, USA.*

MAES, P. (1994): Agents that reduce work and information overload. *Communications of the ACM*, 37:31-40.

MOORE, J. (1995): Participating in Explanatory Dialogues: *Interpreting and Responding to Questions in Context.* MIT Press, Cambridge, Massachusetts.

RAM, A. (1991): Interest-based information filtering and extraction in natural language understanding systems. *Bellcore Workshop on High-Performance Information Filtering.*

RAM, A. (1992): Natural language understanding for information-filtering systems. *Communications of the ACM,* 35(12):80-82.

REITER, E. and DALE, R. (2000): *Building natural language generation systems*. Cambridge University Press.

RICH, C. and SIDNER, C. (1998): Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 3(8):315-350.

SCHIFFRIN, D. (1987): *Discourse Analysis*. Cambridge University Press, England.

SEARLE, J. (1975): A taxonomy of illocutionary acts. In *Language, mind and Knowledge: Minnesota Studies in the Philosophy of Science*, 7. University of Minnesota Press.

SHAPIRO, S. (1995): Sneps 2.3. user's manual. Technical report, Department of Computer Science, SUNY at Buffalo, NY, USA.

TONG, L., CHANGJIE, T., and JIE, Z. (2001): Web document filtering technique based on natural language understanding. *International Journal of Computer Processing of Oriental Languages*, 14(3):279-291.

VAN DIJK, T. (1995): De la gramática del texto al análisis crítico del discurso. *BELIAR, Buenos Aires*, 20(6).

VOORHEES, E. (2000): Overview of trec-9 question-answering track. *The Ninth Text Retrieval Conference (TREC-9), Gaithersburg, USA.*

VOORHEES, E. M. (1999): Natural language processing and information retrieval. In *SCIE*, 32-48.

WRIGHT, H., POESIO, M., and ISARD, S. (1999): Using high level dialogue information for dialogue act recognition using prosodic features. *Proceedings of the ESCA Workshop on Prosody and Dialogue, Eindhoven.*

BIOGRAPHICAL NOTES

Anita Ferreira-Cabrera obtained her B.A and M.A in Linguistics both from Universidad de

Concepción (Chile), and her PhD in Linguistics from Universidad Catolica de Valparaíso (Chile). She is an associate professor at the Department of Linguistics, Universidad de Concepción (Chile). Her main research topics include Natural-Language Processing, Intelligent Tutorial Systems, Intelligent CALL. Presently, she is a PhD candidate in AI at University of Edinburgh, UK.

John Atkinson-Abutridy received his B.Eng and M.Eng (Informatics) from Universidad Técnica Federico Santa María (Chile). At present, he is a PhD candidate in AI at the University of Edinburgh, UK. He is an assistant professor at the Departamento de Informática, Universidad de Concepción (Chile). His research interests include Natural-Language Processing, Knowledge Discovery from Texts, Artificial Intelligence and Evolutionary Computation.

Database Browsing

José Martinez

Institut de Recherche en Informatique de Nantes (IRIN / BaDRI)
Ecole polytechnique de l'Université de Nantes – La Chantrerie – B.P. 50609 – F-44306 Nantes Cedex 3
Tel.: +33 2 40 68 32 56 – Fax: +33 2 40 68 32 32
Email: Jose.Martinez@irin.univ-nantes.fr

In the field of database browsing, there has been much recent effort leading to advances in hypermedia methodologies. Based on this previous work and the projects that we have implemented on this topic, we provide a formal framework to clarify the concepts and functionality of the various systems. The attributes of the proposal are that: (i) it does not intrude into the database schema, (ii) it has been applied to object-oriented as well as relational databases, (iii) it combines the user – and data-centred approaches, and (iv) it offers a fair trade-off between uniformity and customisability.

Categories and subject descriptors: H.4.3 (Information Systems Applications): Communications Applications – information browsers; H.5.4 (Information Interfaces and Presentation): Hypertext/Hypermedia – navigation; D.2.11 (Software Engineering): Software Architectures – domain-specific architectures.

General Terms: Design.

Additional Key Words and Phrases: Hypermedia methodology, customisability, generalised tours, formal framework.

1. INTRODUCTION AND MOTIVATIONS

Hypertext (Conklin, 1987), and more recently hypermedia (Nielsen, 1990; Hardman, Bulterman and van Rossum, 1994) are no longer isolated research domains. A review of the proposed architectures, including the seminal Dexter model (Halasz and Schwartz, 1994), highlights the fact that hypermedia systems need database facilities. Smith and Zdonik (1987) recommend the use of an OODBMS as the more rational solution to store hypertext data. As an example, Hyperwave is a hypermedia system based on an object-oriented database engine (Maurer, 1996).

Our proposal is situated within the field of hyper-bases, i.e., hypermedia presentations of database content, especially on the Internet, such as e-commerce sites, portals, or corporate sites. For that purpose, mature and full-fledged methodologies have emerged such as HDM (*Hypermedia Design Model*) (Garzotto, Paolini and Schwabe, 1993), RMM (*Relationship Management Methodology*) (Isakowitz, Stohr and Balasubramanian, 1995) or OOHDM (*Object-Oriented*

Manuscript received: April 2001
Associate Editors: Michael Schrefl and John Roddick

Hypermedia Design Model) (Schwabe and Rossi, 1995). They often incorporate the complete lifecycle in a homogeneous notation, including schema design, users' modelling, presentation design, human-machine interaction and so on. They are bounded to pre-existing proposals and we contend that several aspects are better addressed by using those pre-existing and often well-established proposals directly. For instance, database schema modelling could be done either with low-level functional, multi-valued and join dependencies, with entity-relationship modelling, with any of the latest object-oriented modelling techniques, or even without any methodology for small databases[1]. Role modelling and the definition of conceptual views are tasks that have already been conducted by the database designer and/or administrator for security and authorisation reasons. In addition, designing human-machine interfaces is a complex task that requires some level of expertise. Indeed, in any information system most of the above issues have been conducted in one way or another. Moreover, each aspect is subject to evolution. Therefore, being able to reuse and combine various state-of-the-art approaches rather than to depend on a proprietary solution is a software engineering goal of its own.

From the practical point of view, thanks to third-party tools, it is possible to implement database-powered Web sites easily. For instance, ASP (*Active Server Pages*) or PHP (*Hypertext Preprocessor*, formerly *Personal Home Page*) are used daily on a vast number of sites. However the produced code does not clearly separate presentation issues, navigation issues, and above all users' roles. In the Internet world, we strongly recommend an XML-based implementation. Let us imagine that we have the perfect tools at our disposal: on the one hand, we can query a database using the forthcoming XQuery language, thus obtaining XML documents, and, on the other hand, we can send the results to a Web browser that can interpret both XML documents and XSLT style sheets. This paper thus addresses what still has to be introduced between the database and the browser. In other words, we direct the reader's attention to the core part of a database-based hypermedia system.

The general idea is to reuse as much as possible the tools and techniques that have been developed so far, then to consider what is still missing to achieve our goal. The objective is to offer the possibility of freely navigating inside databases rather than to query them, something that is unfriendly for most users. Basically, in our proposal we overlook the data model and the presentation model. Therefore, we focus on the core of a hypermedia presentation of a database content and highlight the main feature, which is customisability, i.e., what appears on the screen should be related not only to the database content but also to several other considerations, among which the most important is the connected user.

The reader has to keep in mind that the proposal is a framework that has to be adapted. However, we provide general guidelines, and even formal notations that can be derived into actual code in a straightforward manner. Indeed, this high-level of abstraction is advantageous because it allows several implementations. We shall see that this framework has been implemented three times, in different environments, with different tools. Should the objectives evolve, e.g., WAP applications, the proposed framework should remain essentially the same. In addition, the choice of a formal presentation should help with the comparison of various approaches. The proposals that we know of are based on semi-formal definitions only. Additionally, this approach may well extend to other areas, such as semi-structured documents.

This paper is organised as follows. First, in Section 2, we give a short background on hypermedia and hyper-bases. In the following section, we introduce our framework, part of a

[1] *There is no perfect bijection between relational and entity/relationship modelling (Mannila and Räihä, 1992). Formally, the former is the better.*

general and well-accepted methodology. A few implementation highlights are given in Section 4, differentiating between functional and declarative options. A comparison with several approaches is conducted in the next section. Finally, the conclusion points out an important direction that we want to pursue.

2. BACKGROUND

For our purpose, we distinguish two phases in the history of database-based hypertexts: (i) hypertexts stored within a database and (ii) database content viewed as a hypertext.

The common underlying concepts of hypertexts have been standardised and formalised in the Dexter hypertext reference model (Halasz and Schwartz, 1994), which proposes a three-level architecture: storage, component, and presentation. Basically, a hypertext is a directed graph. The term *component* is the standard name for vertice whereas *link* stands for edge. A component is any information that can be displayed at once (composite components are also taken into account) whereas links allow users to display a related component, hence navigating the hypertext. The link concept has been considered as the most important (Bieber and Isakowitz, 1995). The origin and end of a link are called anchors. They represent a location inside a component, hence one can design inter-component links as well as intra-component links, and they can be crossed in either or both ways. Next, at the storage level, the hypertext objects are accessible through Unique IDentifiers (UIDs) that are similar to Object IDentifiers (OIDs) in an object-oriented database management system (OODBMS). Finally, presentation specifications, to visualise components onto the screen, are mostly left open to the designer.

To avoid *user's disorientation*, a set of presentation guidelines have been designed (Nielsen, 1990). This undesirable phenomenon is limited through the consistent use of maps or indexes for global positioning, style sheets for homogeneous presentations, and tours for navigating. Of special interest is the notion of tour in hypermedia design (Rutledge, Hardman, van Ossenbruggen, and Bulterman, 1998). In fact, more often than not we have to browse not a single object, but a collection of inter-related objects. The most cited kinds of tours are guided and indexed tours. *Guided tours* look like linked lists of pages that can be accessed sequentially through a next button (and optionally a previous one too). *Indexed tours* allow users to access a set of pages more rapidly through an intermediate page, the so-called index, that leads in turn to the individual pages.

This approach has been implemented by Hyperwave (Maurer, 1996), a hypermedia system based on an object-oriented database engine. The main problem with this solution is that the database schema has to be mapped into the offered data structures.

At present, an alternative for implementing hypermedia (Web-based) databases is to translate the database content into hypertext components. This is another complete though specific instantiation of the Dexter model. Basically, object or tuples correspond to nodes, whereas relationships or foreign keys correspond to links. In this area, we reuse mainly ideas from HDM, RMM and OOHDM.

HDM-lite (Garzotto, *et al,* 1993), used by AutoWeb (Fraternali and Paolini, 1998) and its descendant WebML (Ceri, Fraternali and Bongio, 2000), is also based on three levels when designing a Web-based database: a *conceptual hyper-base schema*, an *access schema*, and a *presentation base*. First, a *conceptual hyper-base schema* is designed in the form of an *extended* entity-relationship diagram. The method highlights the aggregation relationships. In addition, a designer can add shortcut relationships directly into the schema. HDM-lite requires the definition of an external (printable) name that will identify the sink object in an anchor, and that is not necessarily a key.

Secondly, accesses into the database are pointed out in an *access schema* that describes both the entry points into the database (*collections*), and the navigation paths between the entities (*traversals*, i.e., tours). The former consists of *entity collections*, which are the extensions of all the classes of interest, and of the *entry collection* that contains pointers to the entity collections. For navigating between instances, a set of collection traversals is offered: guided, indexed, and indexed guided tours, as well as a single page containing all the sink objects. Collections can be ordered as well as constrained by a predicate (a simple conjunctive form of attribute/operator/value is accepted). HDM-lite defines the visibility of collections: either global, or local to an entity, a component, or a set of objects. The characteristics of the traversals are stored into a relational database repository, the *access base(s)*. Several sets of access specifications can be defined in order to accommodate different kinds of users. From these specifications, actual code can be generated in various languages.

Thirdly, the description of the pages is given with *style sheets* for component and traversal pages. These are described in an SGML-like syntax that specifies the placement of graphical objects into a grid and their presentation attributes (WebML uses XML and XSLT instead). Again, this aspect of HDM-lite gives birth to *presentation bases* stored into the relational repository. Some inherent limitations, due to the static description of data in a relational repository are removed by allowing utility components to be inserted into a presentation, i.e., anything that can be understood by a Web browser, e.g., a Java applet for advanced features.

RMM (*Relationship Management Methodology*) (Isakowitz, Stohr and Balasubramanian, 1995; Bieber and Isakowitz, 1995) is also based on entity-relationship modelling. It provides guidelines in order to translate an entity-relationship schema into a hypermedia schema. In short, entry points are defined for entity extensions; guided tours, indexed tours, and a simple combination of them in the form of index guided tours are then proposed for navigating between entities; finally, entity instances should be translated into hypermedia components. The original extension consists of splitting into slices the instances that have too much information to be displayed at once on a single page, leading to several sub-pages connected to a root entry page, e.g., a mix of photographs or drawings for an animal (skeleton, body, head, male/female…), along with long descriptions of its living environment, and so on.

OOHDM (*Object-Oriented Hypermedia Design Model*) (Schwabe and Rossi, 1995), now based on UML (*Unified Modeling Language*) comprises four activities: conceptual modelling, navigational design, abstract interface design, and implementation. It adds some elements to ease the development of hypermedia databases. For instance, union types are allowed, e.g., an attribute can be defined as either an image, or a video. The relevant point of this methodology is to emphasise the fact that an application is a view of the conceptual model. The concept of navigational context is introduced to describe several navigational structures on top of a single conceptual model. The abstract interface design consists of describing a hierarchy of graphical objects (buttons, text fields…) together with the associated behaviour when receiving events.

Contrary to OODHM that uses some proprietary extensions, Baumeister, Koch and Mandel (1999) propose to describe such a database-based hypermedia design by relying solely on standard UML modelling: a class diagram is used for the conceptual model; then stereotypes, OCL (*Object Constraint Language*), and object diagrams are used for each navigational model, a subset of the class diagram; finally, the presentation model is described respectively by composition diagrams and state-charts for the static and dynamic parts of a presentation. This proposal exemplifies that state-of-the-art in hypermedia modelling is continuously evolving and can be incorporated into more general modelling techniques.

3. A FORMAL FRAMEWORK

It is important to remember the idea that leads to this framework. The main objective is to offer the possibility of navigating inside a database through hypermedia facilities. However, we do not want to be bound to either by any particular hypermedia methodology, or any specific tool. Therefore, we overlook the modelling phase and the presentation details. This proposal addresses what has still to be introduced between the database and the browser.

3.1 A Methodology

Roughly speaking, we can draw the following methodology for hypermedia database design. First, we design a data model, using known database methodologies, e.g., E/R or OMT/UML, and above all *ignoring* hypermedia issues. The rationale is that any existing application should benefit from hypermedia extensions without redesigning them.

At the other end, we delegate the presentation work to a dedicated and separate tool. For database applications that are moving to the Web, the right choice is to rely on XML-compliant tools. Using a Web browser that can interpret XML and XSLT allows declarative specifications of all the presentation details, from layout to font colours. It allows even customisation by the end-user.

In between lies the core of a hypermedia system, that we formalise here. The basic idea is to provide a means for *translating "objects" into "pages"*. This observation leads to the major function:

$$\Phi\text{: } \mathit{Instance} \rightarrow \mathit{Page} \tag{1}$$

In view of the fact that end-users continually require the benefits of customised interfaces to an information system (a concept known as "external view" in the database world), we extend the tentative definition:

$$\Phi\text{: } \mathit{Instance} \times \mathit{Role} \rightarrow \mathit{Page} \tag{2}$$

i.e., Φ is a function from an instance to a page, based on a given user's profile, therefore combining the opposing *data and user-centred approaches*. Note that a role can represent either a group of users, or a single user in any database system. In addition, we could accept aliases, i.e., different "roles" for a single person. Besides, this is only one step into the generalisation process. One could add the kind of output device as an additional parameter.

3.2 Defining the Main Function

In order to provide a definition for the function Φ, we have decided to enforce some guidelines in our framework. First, in order *to limit user's disorientation*, it seems important that translations lead to a common presentation for all instances belonging to the same class. This implements some uniformity but only within a particular type of data. (Presentations can still be customised at the class level, by testing the object state, e.g., the names of questionable customers can appear in red capital letters). As a prerequisite to achieving this constraint, let us suppose that the programming language offers a typing function:

$$\mathit{iof} : \mathit{Instance} \rightarrow \mathit{Type} \tag{3}$$

Next, note that Φ (2) turns out to be a multi-polymorph function. It is known that ambiguities arise easily with multi-polymorphism. This is pointless in the context of well-defined mathematical functions, e.g., intersections between geometrical objects (adapted from Date and Darwen, 1998).

However, it is important to avoid confusion in our case where page layouts are not properly constrained, i.e., there is no formal definition of what the output must look like.

As an example, let us suppose we have a class *C* and a subclass *SC* of it. Most users are given access to *SC* instances. We define a subset of users who are allowed a larger view of the database, being able to access all of *C* instances. What to do when these privileged users access *SC* instances is ambiguous. If the first presentation is selected, then the users may simply be unaware that they are seeing instances of *SC* (unless some general presentation is applied to proper data of subclasses). Conversely, the second presentation may hide some attributes to standard users, but privileged users will not be able to view them either.

To remove these ambiguities, we have chosen to emphasise the role argument. This seems natural since role is generally determined at "login-time", and is not modified during the browsing session, whereas instances are changing continuously. Next, for several reasons, among which are authorisation and modularity, it seems wise to avoid arbitrary inter-mixing of any instance with any role. In addition, roles are not necessarily classes in all candidate implementations: they can be provided as names.

With these new guidelines in mind, we can provide a definition for Φ, by using an underlying function and two *sets* of functions:

- one **login manager** associates each user to his or her corresponding directory:
$$m : Role \rightarrow Directory \tag{4}$$
- a **directory** is a *function*, belonging to the *Directory* set, the signature of which is:
$$dj : Type \rightarrow Transducer \tag{5}$$
- a **transducer** is also a *function*, belonging to the *Transducer* set, the signature of which is:
$$tk : Instance \rightarrow Page \tag{6}$$

Note that each **transducer** (6) is a special case of the naive Φ function (1), i.e., each transducer is provided for a specific type/user couple. When ignoring the user's point of view, the simplest way to implement it is in the form of a method (with late-binding) in each class. Various users could still be taken into account through a case list. However, note that besides (object-oriented) schema intrusion, we are back to one of the software engineering problems that lead to object-orientation: hard-wired type selection vs. implicit polymorphism (Meyer, 1986).

Therefore, it is the aim of a **directory** to group together all the transducers that are to be used by a given user's role. By being associated to different directories, two distinct user's roles can either have access to the same transducer, or to a different transducer, *for the same instance*. In other words, each directory achieves a *conceptual view* over the database schema. (From the technical point of view, this is where one of the difficulties lies: a directory is a set of "methods" that are dynamically linked to classes.)

Finally, the **login-manager** is the way to link a user to his or her directory. In function (4), we do not explicitly take into account the classification of users. However, the reader can easily be convinced that standard mechanisms, such as inheritance and/or set operations, can be used to generate specific directories by overriding and/or adding mappings between types and transducers. For instance, graduate students can access the same information as any student, but can also research information such as abstracts, reports, project descriptions, etc. This can be achieved by modifying an "inherited" student directory: (i) adding appropriate transducers for the `abstract`, `report`, and `project` classes, and (ii) changing the mapping from related classes, e.g., adding anchors to the professors' pages to access their research works.

Based on the introduced functions, the formal definition of Φ is:

$$\Phi: \begin{array}{ccc} \mathit{Instance} \times \mathit{Role} & \rightarrow & \mathit{Page} \\ (i, r) & \rightarrow & m(r)(\mathit{iof}(i))(i) \end{array} \tag{7}$$

Let us illustrate it with the example of Figure 1, where arrows represent functional application with parameter(s) at the origin, result at the destination, and the function name in the box. We shall suppose that we are an anonymous user of a Web literature site, willing to access authors' information. At "login-time", the manager associates a given directory to the connected user through *m : Role* → *Directory* (4), which returns the default directory d_1, i.e., *m*(*anonymous*) = d_1. A registered user can have his or her role determined automatically through the identification and authorisation protocol, whereas other users can be associated to a role based on their Internet domain for instance.

Whenever a new instance is to be displayed, the type (name) to which the instance belongs is first retrieved thanks to *iof : Instance* → *Type* (3). Starting from an author's instance, say William Shakespeare, the system retrieves "author", i.e., *iof*(*William Shakespeare*) = *author*. By applying it to the user's directory d_1 : *Type* → *Transducer* (5), the type identifier leads to the corresponding transducer, i.e., d_1(*author*) = $t_{1,1}$, hence a function of both the instance and the role. The transducer can be either a generic transducer or a specific one that is in charge of producing the "page" for a particular class or hierarchy of the application domain.

Finally, the instance is given as an argument to the dynamically retrieved transducer, $t_{1,1}$: *Instance* → *Page* (6), and a page is generated, i.e., $t_{1,1}$(*William Shakespeare*) returns an HTML page containing simple attributes such as name, surname, dates of birth and of death, along with multimedia components, e.g., a picture of the author, a bibliography, *etc*. The page will also contain anchors to provide a way to access related instances such as the list of the books that he wrote. These buttons will activate the same process on the corresponding instance(s) in order to construct another page. Alternatively, related objects could have been (partly) presented within the referenced object, which is quite natural for aggregation relationships.

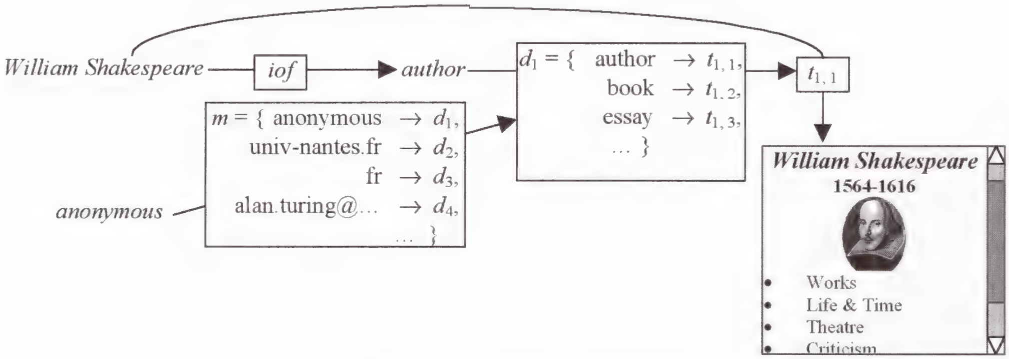

William Shakespeare
iof
author
d₁ = { author → t₁,₁, book → t₁,₂, essay → t₁,₃, … }
m = { anonymous → d₁, univ-nantes.fr → d₂, fr → d₃, alan.turing@… → d₄, … }
anonymous
t₁,₁
William Shakespeare
1564-1616
Works
Life & Time
Theatre
Criticism

Figure 1. An example of applying Φ

3.3 Extension to Tours and Large Objects

It is possible to rely on the sole Φ function for navigating a hyper-base, and more generally any hypermedia graph. However, we have to show how it can take collections into account. Indeed, more often than not we have to browse not a single object, but a collection of objects, especially the entry points into the database, and the database itself is viewed as a collection of entry points. Examples of collection data types are lists, sets, bags, dictionaries, and relations. (A type system is a prerequisite, but we avoid the long and tedious task of defining again similar definitions and direct the reader to such formal definitions, e.g., the O_2 type system (Bancilhon, Delobel and Kannelakis, 1992) or the ODMG (*Object Database Management Group*) standard (Cattel, *et al*, 1997) for OODBMSs, Date and Darwen (1998) for object-relational DBMSs, or even Abiteboul, Hull and Vianu (1995) for relational DBMSs.)

Besides, the following extension is also able to deal with sliced large objects, as proposed by the RMM methodology (Isakowitz, *et al*, 1995). Indeed, this corresponds to a hidden aggregation relationship between the core description of an object and subsets of attributes. Note that in some cases such slicing can be done during the modelling step of the application by using effectively aggregation relationships between a root class and dependent classes. In relational databases the first normal form constraint (1NF) imposes this aggregation for multi-valued attributes.

We deal uniformly with these two cases by introducing a special kind of transducer, the role of which is to generate a *transient* structure: a rooted connected graph. Formally, a **tour transducer** is a function belonging to a subset of the *Transducer* set, i.e., it obeys the signature that has been given for them (6), but the overridden definition is:

$$t_k : \begin{array}{ccc} Instance & \rightarrow & Page \\ i & \rightarrow & \Phi(i'_0) \end{array} \qquad (8)$$

where such a transducer utilises an underlying function, a **tour creator**, defined as follows:

$$\tau_j : \begin{array}{ccc} Instance & \rightarrow & 2^{Instance} \times 2^{Instance \times Instance} \times Instance \\ i & \rightarrow & (X, U, i'_0) \end{array} \qquad (9)$$

where X is any set of instances derived from i, $U \subseteq X \times X$ is a connected graph, and $i'_0 \in X$ is an entry point into the graph. (For the sake of notational simplicity, we rely on pseudo-functional definitions since tour creators have the side-effect of populating the database with transient objects. Where such objects are really created, and when they are destroyed is implementation-dependent.)

In other words, a tour creator translates either a “flat” collection, e.g., a set, into a rooted graph, or a single object into a set of inter-related slices. In fact, a designer can provide any kind of traversal of a collection of objects by taking advantage of various properties that these objects can exhibit, in order to construct on-the-fly a non-existent structure. This result, a local graph, is precisely what the framework is able to navigate… as long as transducers are provided for the generated objects! Therefore, an actual tour (respectively large object) transducer is the combination of a tour creator and of a set of transducers, the recursive definition ending up on transducers that implement an effective translation into a “page”.

This simple extension opens a wide range of possibilities. First, collections can be visited in an *unbounded* variety of ways: the “traditional” guided, indexed, and indexed guided tours, as well as multi-level indexed (Barbeau and Martinez, 1999b), combinations of guided and indexed tours (Barbeau and Martinez, 2000), two-dimensional arrays, lattice tours (Martinez and Loisant, 2002), *etc*. Furthermore, any generated graph can be visited in slightly different ways since we can use different transducers for its nodes, e.g., each page of a guided tour can either contain or not contain pointers to the first and last pages. Finally, since tour transducers form a subset of transducers, they can also be adapted to a given user, e.g., some users may see the list of the titles of Shakespeare's works within Shakespeare's page, whereas another user may see a pointer to a separate page, and still another to a multi-level indexed tour where the works are classified under four categories: comedies, histories, poetry, and tragedies, (indeed several different ways to visit the works can be provided simultaneously).

At this point, performance becomes an issue. Some tours are easy to construct, e.g., the common guided and indexed tours. They are simple variations on the way to visualise the result of a query (written either in SQL, or OQL, or XQuery), i.e., a list of objects or tuples is transformed into a doubly-linked list or a tree. The time complexity is linear, or incurs a small logarithmic penalty for multi-level guided tours. In contrast, other kinds of tours are complex to build. They must be computed off-line and then stored into the database when the number of objects is relatively high. This is the case with Galois' lattices, which have been used for browsing image databases (Martinez and Loisant, 2002), because the best known algorithm to date is quadratic (Godin, Missaoui and Alaoui, 1995). Redundant information, such as materialised views, can help to reduce the complexity and to extend the range of tour variations.

4. ALTERNATIVE IMPLEMENTATIONS

So far, we have been introducing the high-level concepts of our approach (the Φ function and the concept of transducers, i.e., its naive version) and some intermediate level concepts (directories, two kinds of transducers, and the underlying tour creators). They correspond to the analysis and design phases respectively. However, unless every declared function is given a proper definition, there cannot be any implemented version of the framework.

4.1 Functional Definitions

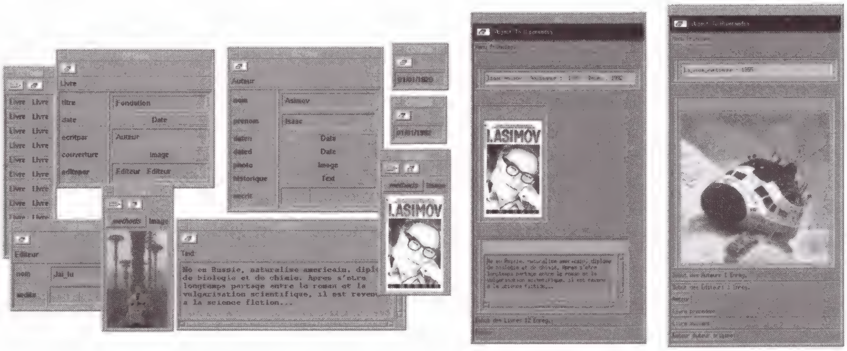
Although relational database management systems (RDBMSs) are currently the industry work-horses, we have opted mainly for object-orientation, hence turning this formal framework into an object-oriented framework (Fayad and Schmidt, 1997; Johnson, 1997). It can be applied to different systems, and it has been (partly) implemented on three occasions. The three implementations were fully functional, hence taking advantage of their knowledge of the underlying database to customise presentations.

The first implementation was achieved within the O_2 OODBMS (Bancilhon, *et al*, 1992), in order to test feasibility, ease of development, and usefulness (Barbeau and Martinez, 1999a). It was simplified due to the absence of uploading. In addition, the presence of a native meta-schema helped to develop a general transducer with default presentation for any type of data (a list of attribute/value pairs and only guided tours). The benefit over the native browser is illustrated in Figure 2 where the novelist Isaac Asimov and one of his books appear in two windows rather than nine. (The O_2 browser opens a new window for each sub-object. Besides, all opened windows are stacked, and must be closed in reverse order.)

Following this, a Java applet was programmed. This choice was motivated by the requirement to present data onto the Internet, coming from distributed databases. In addition, once started, an applet offers some advantages. Firstly, the bandwidth is reduced since only the useful information is sent over the network. Moreover, this may result in improved security, e.g., displayed information cannot be copied easily, or at least accurately, into the local disk. Finally, all the tours are created locally on the client machine, therefore avoiding complex protocols to delete them appropriately as well as server overhead. This version has introduced generalised tours (Barbeau and Martinez, 1999b).

Thirdly, courseware has been developed. It presents different requirements. An important one is that students should have at their disposal a complete collection of HTML pages in order to copy some courses on a floppy disk, and then browse them at home. Consequently, the framework has been applied for generating automatically such pages for printing (a single long page with an introductory table of contents, and internal cross-references), for off-line browsing and overhead projection in class rooms (both with variants on the way the table of contents, the navigation bar, the hierarchical structure, and the key word index behave), and for on-line access to all the electronic courses of the school (with accesses *via* author, topics, department, or year of teaching).

We give the reader an insight into the object-oriented implementation. Barbeau and Martinez (2000) present the corresponding Java classes in more details. The login manager (4) simply corresponds to a map or associative array. (It is somewhat enhanced to accept Internet hierarchical domain names, e.g., to provide a specific view to people connecting from `univ-nantes.fr`.) Similarly, each directory (5) is translated into a map from strings, i.e., the class names, to transducers. Implementing transducers (6) is the actual difficulty. A transducer hierarchy is created. Each class in this hierarchy is a functional class, i.e., a class that contains essentially one method, say apply, and often no instance variable at all. Introducing a new transducer in the system consists in: (i) creating a new subclass in this hierarchy, which contains a new code for the apply method, (ii) creating at least one instance of this class, and (iii) associating this instance to a given class in one or several directories. Tour transducers accept collections of instances rather than single instances and work as indicated in (8) and (9), i.e., transient objects are created and displayed using their corresponding transducers. (In practice, most tours have been optimised, i.e., transient objects are *not* actually created.)



Livre

titre Fondation
date Date
ecritpar Auteur
couverture image
editepar Editeur Editeur

Auteur

nom Asimov
prenom Isaac
dateN Date
dateD Date
photo image
historique Text
ecrit

01/01/1920

01/01/1992

Editeur

nom Jai_lu
edite

Text

Ne en Russie, naturalise americain, dipl
de biologie et de chimie. Apres s'etre
longtemps partage entre le roman et la
vulgarisation scientifique, il est reven
a la science fiction...

(a) (b)

Figure 2. The standard presentation with the O_2 browser (a) vs. the corresponding hypermedia presentation (b) of an author and a book

4.2 Declarative Specifications

It is generally easier to deal with data rather than code, though this approach somewhat limits extensibility. We can illustrate this declarative approach thanks to a medical application. A relational database stores information about patients and practitioners along with medical images of the former and textual and audio annotations of the latter.

Firstly, an instance name is either "DB" for accessing the home page of the system, or a capitalised relation name for accessing several tuples of the corresponding relation, or a pair consisting of a relation name and key value for accessing a specific information, e.g., a patient record.

Secondly, a user identifies him or herself at login-time. He or she can be either a patient with limited access rights, or a secretary who can access all the information except the medical annotations, or a doctor who can access all the information that is related to one of his or her patients.

Then, in the relational world, the multi-polymorphism of Φ (2) is implemented easily in the form of a relation, the key of which consists of a name (either DB, or a relation name, possibly capitalised) and a user's role. The associated attributes consist in an XQuery query and an XSLT style sheet. (Of course, the former will translate into one or a series of SQL queries in the short term.) If we do not consider tour creators, then one execution of Φ is limited to executing the query associated to the connected user and the referenced instance and sending the result along with the corresponding style sheet to the browser, i.e., Φ is implemented as a basic "`select query, sheet from Phi where user = connected_user and instance = referenced_instance`" to retrieve the query and the style sheet, followed by the execution of query. This is planned to be executed on the Web server part of the three-tier architecture. Note that the medical database need not be the same as the database that contains the Φ relation, the queries and the style sheets.

These declarative specifications offer some advantages. First, both the queries and the style sheets use declarative languages; these languages are powerful and allow a lot of work to be done without relying on advanced programming skills. In addition, it is quite simple to make changes to the Φ relation. Changing queries or style sheets is almost immediate. In the last case, the look-and-feel of the application can be modified independently of any other consideration. Adding new roles can either be done from scratch, or they can be derived from existing ones in the form of SQL scripts. Finally, they can be modified even at run-time [2].

Other kinds of declarative specifications can be conceived, such as those used by AutoWeb and WebML. They are more constrained but can be derived into actual code, e.g., ASP scripts.

5. PREVIOUS RELATED WORK

In the two preceding sections, we have introduced a formal framework and detailed the rationales of its design. The overall proposal is based on a three-tier architecture, like the Dexter model. The major difference is that stored nodes and links consist of the database schema of the application. In addition, we and others, like RMM and Araneus (Atzeni, Mecca and Merialdo, 1997), recommend not to be intrusive in the database schema, i.e., we do not impose any extension, restriction, or data model on the database schema design, hence being usable even for old applications without reengineering needs. The database usage is severely limited when it is not fully independent of the hypermedia system, like HDM-lite and OOHDM. HyperWave, especially, provides only a limited number of concepts: directed acyclic graphs, sequences, and clusters (Maurer, 1996). Therefore, the database schema has to be seriously modified in order to fit this restricted number of concepts.

We have argued that any well-accepted database design methodology can be used, as well as corresponding data models, i.e., either object-oriented, or relational, as exemplified in our implementations. However, we advocate an object-oriented model, as ODMG (Cattel, *et al*, 1997),

[2] *This can be obtained, though painfully, from an object-oriented implementation with parameterised classes. For instance, our implemented guided indexed tours can be uni- or bi-directional, contain or not direct pointers to the first or last page, contain or not pointers to the parent page (except for the last one that always incorporates it). Therefore, it would be possible to create an extent of named guided tours, and then ask transducers to retrieve and clone these guided tours rather than to create new ones ex nihilo whenever they are called.*

because it can provide both stored and calculated attributes and relationships [3], including the short-cuts of HDM-lite, as a uniform interface to the objects. It would be even better if operations could be added on top of existing and already instantiated interfaces.

From the presentation point of view, we can totally change the look-and-feel of the presentation without sacrificing the overall uniformity that is supported by the framework. Even the end-user can do it when using XML. In this way, final presentation becomes a less significant concern, contrary to HyperDesign which focuses on user-interface design (Balasubramanian and Turoff, 1995).

Therefore, we put the emphasis on the browsing capabilities. The Φ function (2) combines successfully the somewhat opposite data and user-centred approaches. In contrast, WSDM (*Web Site Design Method*) (De Troyer and Leune, 1998), emphasises the user-centred approach. It starts by identifying and modelling the potential users of a hypermedia system. Our solution has been mentioned in sections 3.2 and 4.2.

A limited set of collection tours is offered by other methodologies, especially PESTO that has only guided tours (Carey, Haas, Maganty and Williams, 1996) (but adds some querying capabilities). Some authors claim extensibility. Our framework goes farther by giving a generic way to extend tour capabilities. This feature decreases considerably the user's concentration, and increases the freedom to move around the objects of a database. In addition, this extension applies directly to the slice concept of RMM. Note that EORM (*Enhanced Object-Relationship Model*) (Lange, 1996) puts emphasis on the representation of links as full objects. In a way, this is comparable to our generalised tours that most often provide a means to browsing relationships between entities, i.e., links between components.

Last but not least, it unifies and subsumes the main previous methodologies under a single formalism. The originality of RMM, namely slices, has been mixed with tours into a single concept. The key concept of views corresponds to directories. It has been identified by OOHDM, translated into access bases in HDM-lite, or composition and navigation models in WebML. Finally, HDM-lite provides explicitly visible collections. For local visibility, each transducer can do the job. For global visibility, it is in practice a bootstrap problem. When the user connects to the database, he or she has first to access the meta-database, and the corresponding transducer for this initial object can filter the authorised collections of objects. Furthermore, the transducer instantiations can be based both on code and data. As a trade-off between these two approaches, we advocate as much as possible to incorporate database queries.

Therefore, the only difference is from the implementation point of view: we do not offer a full CASE tool like AutoWeb or Araneus, which generate automatically relational database-based Web applications, but a corresponding framework that has to be instantiated.

6. CONCLUSION AND FUTURE WORK

Offering several ways to browse sets of data is extremely important for end-users who are navigating into a large and complex database (or a set of loosely connected heterogeneous databases). In effect, in addition to limiting the disorientation problem, this also diminishes the cognitive overhead of having to remember several paths that lead to parts of the final result. In addition, this postpones the moment when the user will have to start querying the database rather than browsing it, something that is unfriendly for casual and/or naive end-users.

We proposed a formal framework that provides a common background for alternative implementations. It lies between a three-tiered architecture without being intrusive on the storage

[3] *This is terminology of the ODMG standard.*

level, nor even the data model. It enforces some rules of good practice, balancing uniformity and customisability. It also offers a trade-off between data-centred and user-centred approaches. At the implementation level, it allows to combine declarative and functional programming.

For the future, we think that it is possible to build on this framework in order to cope with new requirements of hyper-bases. One direction being currently pursued is to create interactively, at run-time, complex tours as combinations of simpler tours. This would lead to a powerful combination of browsing and querying capabilities. In addition, an industrial collaboration has been started with France Télecom Formation, the goal of which is to reorganise their numerous courses, provide a standard platform for storing the courses, and allow multi-device distribution of the courses. This will be an ideal application for testing the extension of this framework to semi-structured data.

ACKNOWLEDGEMENTS

We would like to acknowledge the implementation efforts of several of our students: Cyril Tiollier, and more particularly Pejman Parandi for the O_2 implementation that started this project; Franck Barbeau for the Java object-oriented framework developed during his Ph.D. studies; Catherine Lebrun and Erwan Loisant for the C++ and PHP developments of a courseware; as well as Jean-Christophe Guiotte and Jacques Raynard for a started medical application that is going to take advantage as much as possible of the XML *lingua franca* and related tools.

REFERENCES

ABITEBOUL, S., HULL, R., and VIANU, V. (1995): Foundations of databases. Addison-Wesley.

ATZENI, P. MECCA, G. and MERIALDO, P. (1997): To weave the web. Proc. of the 23rd Conference on Very Large Databases (VLDB'97), Athens, Greece, 206-215.

BALASUBRAMANIAN, V., and TUROFF, M. (1995): A Systematic approach to user interface design for hypertext systems. Proc. of the 28th Hawaii International Conference on System Sciences, Vol. III, IEEE CS Press, Los Alamitos, California, 241-250.

BANCILHON, F., DELOBEL, C., and KANNELAKIS, P. (eds) (1992): Building an object-oriented database system: The Story of O_2. Morgan-Kaufmann.

BARBEAU, F., and MARTINEZ, J. (1999a): OTHY: Object to hypermedia. Proc. of the 11th Conference on Advanced Information Systems Engineering (CAiSE*99), Heidelberg, Germany, 349-363.

BARBEAU, F., and MARTINEZ, J. (1999b): About tours in the OTHY hypermedia design. Proc. of the 5th International Computer Science Conference: Internet Applications (ICSC'99), Hong Kong, China, 146-155.

BARBEAU, F., and MARTINEZ, J. (2000): How to visit data with OTHY. Proc. of the 15th Annual Symposium on Applied Computing (SAC'00), Como, Italy, March 19-20, 909-914.

BAUMEISTER, H., KOCH, N., and MANDEL, L. (1999): Towards a UML extension for hypermedia design. Proc. of the 2nd International Conference on the Unified Modeling Language – Beyond the Standard (UML'99), Fort Collins, Colorado, p 614-629, October 28-30 (in LNCS 1723).

BIEBER, M., and ISAKOWITZ, T. (1995): Designing hypermedia applications. Communications of the ACM, 38(8):26-29.

CAREY, M., HAAS, L., MAGANTY, V., and WILLIAMS, J. (1996): PESTO: An integrated query/browser for object databases. Proc. of the 22nd International Conference On Very Large Data Bases (VLDB'96), Mumbai (Bombay), India, 203-214.

CATTEL, R. G. G., BARRY, D. K., BARTELS, D., BERLER, M. D., EASTMAN, J., GAMERMAN, S., JORDAN, D., SPRINGER, A., STRICKLAND, H., and WADE, D. E. (1997): The object database standard: ODMG 2.0. Morgan Kaufmann.

CERI, S., FRATERNALI, P., and BONGIO, A. (2000): Web modeling language (WebML): A modeling language for designing web Sites. Proc. of the 9th International World Wide Web Conference (WWW'00), Amsterdam, Netherlands.

CONKLIN, J. (1987): Hypertext: An introduction and survey. IEEE Computer, 20(9):17-41.

DATE, C. J., and DARWEN, H. (1998): Foundation for object/relational databases : The Third Manifesto. Addison-Wesley.

DE TROYER, O. M. F., and LEUNE, C. J. (1998): WSDM: A User-centered Design Method for Web Sites. Proc. of the 7th International World Wide Web Conference (WWW7), Brisbane, Australia.

FAYAD, M. E., and SCHMIDT, D. C. (1997): Object-oriented application frameworks. Communications of the ACM, 40(10):32-38.

FRATERNALI, P., and PAOLINI, P. (1998): A conceptual model and a tool environment for developing more scalable, dynamic, and customizable web applications. Proc. of the 6th International Conference on Extending Database Technology (EDBT'98), Valencia, Spain, 421-435.

GARZOTTO, F., PAOLINI, P., and SCHWABE, D. (1993): HDM – A model-based approach to hypertext application design. ACM Transaction on Information Systems, 11(1):1-26.

GODIN, R., MISSAOUI, R., and ALAOUI, H. (1995): Incremental concept formation algorithms based on galois (concept) lattices. Computational Intelligence, 11(2):246-267.

HALASZ, F., and SCHWARTZ, M. (1994): The Dexter hypertext reference model. Communications of the ACM, 37(2):30-39.

HARDMAN, L., BULTERMAN, D. C. A., and VAN ROSSUM, G. (1994): The Amsterdam hypermedia model: adding time and context to the Dexter model. Communications of the ACM, 37(2):50-62.

ISAKOWITZ, T., STOHR, E., and BALASUBRAMANIAN, P. (1995): RMM: A methodology for structured hypermedia design. Communications of the ACM, 38(8):34-44.

JOHNSON, R. E. (1997): Frameworks = (Components + patterns). Communications of the ACM, 40(10):39-42.

LANGE, D. (1996): Object-oriented hypermodeling of hypertext-supported information systems. Journal of Organizational Computing and Electronic Commerce, p 269-293, Fall.

MANNILA, H., and RÄIHÄ, K.-J. (1992): The design of relational databases. Addison-Wesley, 318

MARTINEZ, J., and LOISANT, E. (2002): Browsing image databases with Galois' lattices. Proc. of the 17th Annual Symposium on Applied Computing (SAC'02), Madrid, Spain, March 11-14, 791-795.

MAURER, H. (ed) (1996): HyperG is now HyperWave: The next generation web solution. Addison-Wesley.

MEYER, B. (1986): Genericity versus Inheritance. Proc. of the 1st Int'l Conf. On Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'86), Portland, Oregon, September, 391-405.

NIELSEN, J. (1990): HyperText and HyperMedia. Academic Press.

ROSSI, G., SCHWABE, D., and LYARDET, F. (2000): Abstraction and reuse mechanisms in web application models. Proc. of ER'2000 ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and the World Wide Web and Conceptual Modeling, Salt Lake City, Utah, 76-88.

RUTLEDGE, L., HARDMAN, L., VAN OSSENBRUGGEN, J., and BULTERMAN, D. C. A. (1998): Structural distinctions between hypermedia storage and presentation. Proc. of the 6th ACM International Multimedia Conference (ACM MM'98), Bristol, UK.

SCHWABE, D., and ROSSI, G. (1995): The object-oriented hypermedia design model. Communications of the ACM, 38(8):45-46.

SMITH, K. E., and ZDONIK, S. B. (1987): InterMedia: A case study of the differences between relational and object-oriented database systems. Proc. of the International Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'87), Orlando, Florida.

BIOGRAPHICAL NOTES

José Martinez received his Ph.D from the University of Montpellier II, France, in December of 1992, with designing new recovery and concurrency control protocols in object-oriented databases. Since, he has been an associate professor at the Polytechnic School of the University of Nantes, France, in the computer engineering department, as well as a researcher in (multimedia) search-by-content, indexing, intelligent retrieval and browsing. He obtained his D.Sc. in March of 2002.

SAWT: A New System for Secure and Anonymous Web Transactions over the Internet

Changjie Wang, Fangguo Zhang and Yumin Wang
P.O.Box 119, National Key Lab. on ISN,
Xidian University, Xi'an 710071, P.R.China
Email: cjwang@ee.cityu.edu.hk , ymwang@xidian.edu.cn

This paper proposes a new kind of Secure Anonymous Web Transaction (SAWT) system for anonymous browsing and communication on the Web with high security. In the proposed system, normal users can surf or shop online anonymously while malicious accesses to a Web server can be traced and discovered. The latter property has not been achieved in other existing systems, which can bring greater fairness for both users and Web servers.

Keywords: Anonymity, Proxy Server, Group Signature, and Elliptic Curve

1. INTRODUCTION

With the development of computer science and the popularisation of the Internet, both private communication and commercial transactions on the World-Wide-Web are becoming important and has attracted much research interest. Many of the security concerns for existing systems focus on eavesdropping to prevent outsiders from listening in on electronic conversations. Encryption of communication to and from web servers can effectively hide the content of a conversation from eavesdroppers, and has been integrated into many systems. However, the hiding of the identities of users is often not considered. Thus, eavesdroppers can still learn information such as the IP addresses of users and server computers, the length of the data being exchanged and the time and frequency of these exchanges. Encryption also does little to protect the privacy of the user from the server. It is easy for a Web server to record the contents of each access by checking the log file, which contains much information about its visitors. The server can also record other information such as the user's IP address, Internet domain name, workplace, approximate location and the type of computing platform being used. With additional effort, this information can be combined with other data to invade user's privacy. This is analogous to being asked to register private information when roaming in shops or parks in the real world.

Some proposals have been suggested for hiding the identities of users in an electronic transaction, such as blind signature scheme (Chaum, 1983), and steganographic techniques (Neil, Zoran and Sushil, 2000). Most of those proposals are suitable for offline electronic transactions, where multiple iterations are commonly needed, and are short of anonymity control mechanism, which limits their usage in online Web transaction applications. There are also some existing systems to protect user's anonymity on the Internet, such as Onion Routing (Reed, Syverson and Goldschlag, 1998), Anonymizer (Anonymizer, 2002), LPWA (Gabber, Gibbons, Marias, and Mayer,

Manuscript received: April 2001
Associate Editors: Michael Schrefl and John Roddick

1997) and Crowd (Michael and Rubin, 1998; Michael and Rubin, 1999), but there still exist some deficiencies in these systems, as will be discussed in next section. In this paper, we propose a new kind of Secure Anonymous Web Transaction (SAWT) system to increase privacy of Web transactions. Compared with other existing anonymous systems, one important and attractive feature of the SAWT system is that both the anonymity of normal users and the discovery and tracing of malicious access are achieved. An improved group signature scheme is adopted for this purpose. In the SAWT system, a user's URL requests are submitted by a series of proxy servers and attached with the user's signature. The signature can be verified by anyone but may be "opened", i.e. to find the signer, only by a special group manager. Thus, the identity of users is protected unless the access is deemed to be hostile.

This paper is organised as follows. In Section 2, we briefly introduce some typically existing anonymous systems, their properties and deficiencies. We also present the properties of the SAWT system compared with other existing systems. An overview of the SAWT system is presented in Section 3. Section 4 discusses the procedure, key technique and protocols of the SAWT system in details. Simulation results and a performance comparison of SAWT and Crowds systems are presented in Section 5 and finally the conclusion.

2. RELATED WORK

Many solutions and systems have been proposed for hiding the identity of users in electronic communication over the Internet, such as anonymous email system, anonymous Web browsing system, anonymous connection system and etc. We only discuss the anonymous connection and browsing systems in this paper, since they are more closely related to our system.

We first discuss anonymous connection systems. The basic and widely used solution for anonymous communication was suggested by Chaum (1981), where a basic block called mix is used to route the data among the communication parties. The *mixes* may reorder, delay, and pad traffic to complicate traffic analyses so that the correspondences between messages in its input and output can be hidden. In this way, an anonymous connection over the Internet can be set up. The Onion Routing system (Reed, Syverson and Goldschlag, 1998) is another well-known solution for anonymous connection over the Internet, in which the onion routers used are based on the mixes. There are a number of onion routers in the Onion Routing system, which have the functionality of ordinary routers, combined with mixing properties. The system works in the following way: The initiating application, instead of making a connection directly to a responding server, makes a connection to an application-specific "onion routing proxy" on some remote machine. That proxy then defines a route and builds an anonymous connection through the onion routing network (i.e. several onion routers between the source and destination) by constructing a layered data structure called an onion and sending that onion through the onion routing network. Any intermediate onion router that receives an onion peels off its layer (note that each layer of the onion define the next hop in a route), reads from that layer the name of the next hop and the cryptographic information associated with that hop in the anonymous connection, pads the embedded onion to some constant size, and sends it to the next onion router. In this way, each onion router can only identify adjacent onion routers along the route and the data passed along the route appears different at each router. When there are many other anonymous connections existing in the same time (this is the real case in practice), observers will only see message flowing through the onion network, but cannot tell who is communicating with whom, as the data appears different at each onion router. Such anonymous systems are designed mainly for the protection of the unlinkability of source and destination to resist the traffic analysis attack over the Internet. One problem of those systems is that the anonymity of the source and the destination is less considered. That is, the identity of the sender and the receiver has to be revealed to the first and the last onion router separately.

In practice, the protection users' privacy while surfing the Web is another sensitive problem in private communication over the Internet. Some systems and tools (Gabber, *et al*, 1997; Michael, *et al*, 1998; Michael, *et al*, 1999; Kristol, Gabber, Gibbons, Matias, and Mayer, 1999) have been developed for this purpose. One of the best-known anonymity tools is Anonymizer (2002), which is a Web site that serves as a simple proxy for Web requests. In this system, every user's requests for a URL are submitted by Anonymizer instead of the user himself. When receiving a reply from the server, Anonymizer will send the response back to the user's browser. Since the only IP address revealed to the Web site is that of the Anonymizer, the system can protect the anonymity of users from end servers. Some other systems also have similar mechanisms, such as LPWA (Lucent Personalised Web Assistant). One disadvantage of such systems is that users are not anonymous to Anonymizer itself, thus complete anonymity is not obtained and the anonymity of all the users will be compromised if the "single point" – Anonymizer is attacked.

Crowds is a more cogent anonymous browsing system proposed by Reiter and Rubin (Reiter, *et al*, 1998). In Crowds, no single proxy server is required. Instead, this system is based on the idea that people are anonymous when blending into a crowd. The crowd is made up of some users running the Crowds software and the Crowds users forward HTTP requests to a randomly selected member of the crowd. Neither the end server nor any crowd member can determine the origin of requests. In this way, complete anonymity for users is achieved. However, such property leads to a problem. If one user makes a malicious access to a Web server through Crowds, such hostile action will not be linked to the user's real IP address due to the complete anonymity of Crowds.

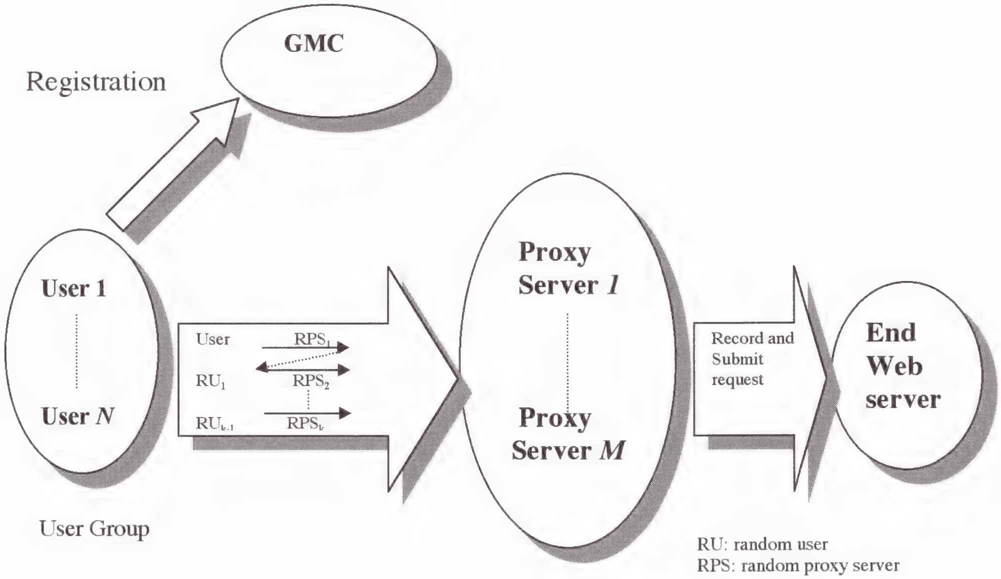
In order to overcome the deficiencies above, we propose a new Secure and Anonymous Web Transaction (SAWT) system in this paper, which combines the techniques of both Anonymizer and Crowds systems. In addition, an improved group signature scheme is adopted for tracing of the malicious connection. Compared with the existing anonymous system, the SAWT system has the following advantages:

- The source anonymity is protected. That is, the identity of the surfer is hidden while browsing online. This property is less considered in *mix* and Onion Routing system.
- The complete anonymity of the normal web surfers is achieved. In the SAWT system, the users' anonymity is not dependent on the security of any single proxy server. This means that the identity of the user will be anonymous even if one or more proxy servers along the route are compromised. This property can solve the security problem in the Anonymizer system.
- Malicious connection tracing is another important feature of the SAWT system. Any malicious accesses to a Web server through the SAWT system can be traced and discovered later. This feature has not been achieved in other existing systems.

3. OVERVIEW OF THE SAWT SYSTEM

This section presents an overview of *Secure and Anonymous Web Transactions (SAWT)* system. We first start with a simplified model, then introduce the working procedure.

The SAWT system can be represented simply by a model with four participators, which are "all users", a believed third party, a series of proxy servers and an end Web server. Here, all users form a group and are controlled by the believed third party called GMC (Group Manage Center). There are several separated proxy servers (PS) in a SAWT system (note that only one proxy server is used in Anonymizer system). In a SAWT system, a user's URL request is submitted by a random PS to the end Web server in a similar way to that in Crowds (Michael, *et al*, 1998). At the same time, a signature conjunct with the request will be recorded in that PS's log file. Figure 1 shows the model of a SAWT system:



RU: random user
RPS: random proxy server

Figure 1: The Model of SAWT System

The working procedures of the SAWT system are described as follows

(1) Registration: All the users in a SAWT system will be regarded as a group. To execute Web transaction, a user should register with GMC first. This means that a user should establish a relationship of his/her identity and pseudonym (maybe a number) to GMC, so GMC can revoke the user's anonymity when necessary. Also, the user will get a certificate issued by the GMC that proves registration.

(2) Path establishment: In a SAWT system, a user's URL request is submitted by one of the PS. To make the user anonymous from PS and other users, we propose an approach similar to Crowds system. A user is represented in the group by software, called the *Web transactor*, running on the local computer, which can be obtained from GMC when registration. Once started, the Web transactor first gets the status of every PS from GMC, and then deals with all requests from and to the user's browser. Thus, a request coming from the browser will be sent to the Web transactor directly. When receiving the first request from the local user's browser, the Web transactor initiates the establishment of a random path to the intended Web server. More precisely, as shown in Figure 1, a user's request will be forwarded to a PS randomly selected by the Web transactor. When this PS receives the request, it flips a biased coin to determine whether to submit the request to the end Web server or just forward it to another user randomly selected from the user group. This procedure is repeated until a PS determines to submit the request to the end Web server (Such procedure is repeated *k* times in Figure 1). Thus, the path establishment is finished. Note that both users and PS can remember every path by appointing a number to each path.

(3) Signature: In this step, when a PS (RPS_k in Figure 1) determines to submit the request to the end Web server, data about this request is produced by that PS and sent to the user who initiated this request along the path established. The Web transactor running on the initiator's computer will check the data, and make a signature of the data using the private key generated by himself and the certificate issued by GMC (note that the relevant information of the user's private key has been registered to GMC). Then the signature data will be returned back to the PS through the same path. It is worth mentioning that an improved group signature scheme is used in signature procedure, which will be discussed in detail in Section 4.

(4) Submission: The last PS will validate the signature data from the initial user using the group public key publicised by GMC, then submit the request to the end Web server if the validation is successful. The reply from the end Web server will be sent to the initiator by the PS through the same path. It should be pointed out that during this procedure the PS cannot identify the initiator since the request may have been forwarded by several users and other PS. However, it can be guaranteed that the initiator is in the user group, which means that the initiator has been registered with GMC.

(5) Tracing: In case of malicious connection, the end Web server will contact the PS who submits the request. This is achieved by checking the log file to get the connection time, IP address and other information of the PS. Then, the PS will check its record to get the signature data corresponding to the connection and submit the data to the GMC. Thus the hostile user (IP address or some other identity information) can be revealed with the help of GMC.

4. SAWT SYSTEM

In this section, we introduce the SAWT system in detail. We first give the system assumptions and definitions, then discuss the implementing protocols of a SAWT system and finally present a security analysis.

4.1 System Assumptions and Definitions

In our system, an improved group signature scheme is adopted, which is more efficient. The concept of group signature was proposed by Chaum and Van Hegst (1991). In a group signature scheme, each member of an arbitrary large group is allowed to sign messages on behalf of the whole group, and the signatures can be verified using a single group public key. No one but the unique designated group manager can open the signature to find the signer. Thus, the anonymity of the signer is protected.

To date, Cam97 (Jan and Markus, 1997) has been one of the most efficient group signature schemes. It is based on the techniques of the signatures of proof of knowledge of double discrete logarithm (SKLOGLOG) and the signatures of proof of knowledge of the *e*-th root of discrete logarithm (SKROOTLOG). The discrete logarithms used in these two kinds of signatures are in ordinary multiplicative groups. The amount of data transferred is large and the signatures are too long. In the SAWT system, we extend the above two schemes to the elliptic curves so that both the amount of data and the length of signatures are greatly decreased, which leads to improved efficiency and sequentially, wider applications in the Internet. In the following, we give the system assumption and the definition of the SKLOGLOG and SKROOTLOG based on elliptic curves (more details about SKLOGLOG and SKROOTLOG can be founded elsewhere (Jan, *et al*, 1997)).

Assumption: let p be a large prime, $A,B \epsilon GF(p)$ satisfy $4A^3 + 27B^2 \neq 0$. The elliptic curve $E_{(A,B)}\ (GF(p))$ is defined to be the set of points $(x,y) \epsilon\ GF(p) \times GF(p)$ satisfying equation

$y^2 = x^3 + Ax + B$ and a special point O (called infinity). These points form an Abelian group. G is an element of $E_{(A,B)}\ (GF(p))$ with order q, where q is a prime at least 160 bit in length. $R_x(D)$ is the *x-coordinate* of point D (More detailed description of elliptic curves can be found elsewhere (Koblitz, 1987; Miller, 1985)). Let H be a one way hash function, H: $\{0,1\}^* \rightarrow \{0,1\}^k$ ($k \approx 160$), $(n,\ e)$ be a RSA public key pair, a be a specified element of Z_n^* with large multiplicative order modulo both factors of n. SKLOGLOG and SKROOTLOG with base group $E_{(A,B)}\ (GF(p))$ are described as $Q = a^xG$ and $Q = a^eG$ respectively. Denoted by the $c[i]$ i-th rightmost bit of a string c, and S_l the first l bits of S, where l is a security parameter satisfied with $l<k$. Define $(\bullet\|\bullet)$ as the concatenation of two strings.

Definition 1. Define the scheme including the following signing and verifying procedures as **ESKLOGLOG (SKLOGLOG based on elliptic curves) signature**, denoted by *ESKLOGLOG* $[\ x : Q = a^x\ G\ ]\ (m)$

Where, x is the secret key of the user. G, a, and $Q = a^x\ G$ are the corresponding public key. The signing and verifying procedures are described as follows:

Signing: Given the plain message M, the signer first select $b_i \in Z_p$, i=1,...,l, and calculate

$$t_i = a^{b_i}\ G$$

$$c = H\left(M \| R_x(Q) \| R_x(G) \| a \| R_x(t_1) \| R_x(t_2) \cdots \| R_x(t_l)\right)$$

$$S_i = b_i - c[i] * x$$

Signature data includes $(c, S_1, S_2 \ldots\ldots S_l)M$.

Verification: validate the follow equation

$$c = H\left(M \| R_x(Q) \| R_x(G) \| a \| R_x(t_1^{'}) \| R_x(t_2^{'}) \cdots \| R_x(t_l^{'})\right)$$

$$\text{where}\quad R_x(t_i^{'}) = \begin{cases} R_x(a^{S_i}G) & if\ c[i] = 0 \\ R_x(a^{S_i}Q) & otherwise \end{cases}$$

Definition 2: Define the scheme including the following signing and verifying procedures as **ESKROOTLOG (SKROOTLOG based on elliptic curves) signature**, denoted by *ESKROOTLOG* $[\ x : Q = a^e\ G\ ]\ (m)$

Where, x is the secret key of the user. G, e and $Q = x^e\ G$ are the corresponding public key. The signing and verifying procedures are described as follows

Signing: Given the plain message M, the signer first select $b_i \in Z_p$, i=1,...l, and calculate

$$t_i = b_i^e\ G$$

$$c = H\left(M \| R_x(Q) \| R_x(G) \| a \| R_x(t_1) \| R_x(t_2) \cdots \| R_x(t_l)\right)$$

$$S_i = \begin{cases} b_i & if\ c[i]=0 \\ b_i / x & otherwise \end{cases}$$

Signature data includes $(c,S_1, S_2........S_l)M$.

Verification: validate the follow equation

$$c = H\left(M \| R_x(Q) \| R_x(G) \| a \| R_x(t_1^{'}) \| R_x(t_2^{'}) \cdots \| R_x(t_l^{'}) \right)$$

where $R_x(t_i^{'}) = \begin{cases} R_x(S_i^{\ e} G) & if\ c[i]=0 \\ R_x(S_i^{\ e} Q) & otherwise \end{cases}$

4.2 Protocols of Implementation in the SAWT System

In this subsection, we present the protocols of implementation in the SAWT system using the improved group signature scheme described above.

i. GMC initialisation

GMC initialisation is the first step after the whole system is activated. In this step, GMC will generate several public parameters of the whole system

- An RSA public key pair (n,e);
- An elliptic curve $E_{(AB)}(GF(p))$ over field $GF(p)$, a large prime P and an element G of $E_{(AB)}(GF(p))$ with order q, where q is at least 160 bits in length;
- A specified element a of Z_n^* with large order modulo both factors of n;
- An up bound λ of the length of the keys.

After initialistion, GMC will publicise the group public key of the users $\Omega = (n,e,a,E_{(AB)}(GF(p)),G,q,\lambda)$, which may be refreshed over the GMC homepage.

ii. User registration

In a SAWT system, all users are regarded as a large group. Each user should register with GMC before using the anonymous service, he should first register with GMC. The procedure is described as below

User		**GMC**
Select $x \in_R Z_{2^\lambda - 1}$		
Calculate: $Y = a^x$, $Z = (Y \bmod q)\ G$	$\xrightarrow{Y,\ Z}$	record Y, Z and calculate: $V = (Y+1)^d \bmod n$ (Here d is the secret key of the GMC, corresponding to above RSA key pair)
record V	$\xleftarrow{V}$	return V to user

Users should carry out the above procedure when installing the software Web transactor, which can be download from the homepage of GMC. After registration, GMC possesses a suit of data including $\{V, Z, Y\}$(this data should be linked with the user's real identity, such as IP address) and adds this to its database. A registered user gets a suit of data including $\{x, V, Z, Y\}$, where x is the user's private key.

iii. Signing procedure

As described in Section 3, a random path should be established before a user's URL request is submitted to the end Web server,. Then, a signature data about the user's URL request will be recorded by the last PS on the random path. The detail of the procedure is described below:

User		**Proxy server**
	Path establishment ⋮ $\xleftarrow{\quad M \quad}$	Generate a data *M*, including the intending IP address, timestamp, a random number and other information of the request.
Make a group signature of *M* using key pair (x, V), as shown below $M(G',Z',S_1',S_2')$, Where $G' = rG$, $Z' = YG'$ $r \in_R Z_p^*$ $S_1' = ESKLOGLOG\ [\, x : Z' = a^x\, G' \,]\ (M)$ $S_2' = ESKROOTLOG\ [\, V : Z'\, G' = V^e\, G' \,]\ (M)$ Return the signature to PS:		
	$\xrightarrow{M(G',Z',S_1',S_2')}$	Verify user's group signature $M(G',Z',S_1',S_2')$ on M. If valid, submit the request to the end Web server. Also, the signature data will be recorded by the PS.

In this step, every URL request initialed by any user of the group will be submitted to the intending Web server while reserving a group signature data in the log file of one certain PS. Since the PS can not link the user's identity with the signature data, the complete anonymous is achieved. However, the signature data may be checked by GMC to trace the user's IP address with the corresponding registration information if the access is regarded as malicious, as described in 4.2-iv.

iv. Tracing

When receiving a signature data (including $M(G',Z',S_1',S_2')$) corresponding to a suspicious access, GMC should find a Y' from all registration information in its database, which satisfies the equation $Y'G' = Z'$, and Y' can link to the user's real identity (maybe IP address). This procedure is relatively simple and easy to carry out.

4.3 Security Analysis of the SAWT System

In the following, we analyse the security of the system against several threats to user's anonymity.

i. Threats from end Web servers

In a SAWT system, user's anonymity from end Web servers is easy to achieve. As pointed out in Section 3, every request will pass through several PS before reaching the end Web server. Thus, end Web servers only know the IP address of the last PS on the path, but nothing about user's identity.

ii. Threats from middle nodes

As explained above, in a SAWT system, a random path should be established before any request submission. Since the path initiator selects PS randomly when creating the path (see Section 3), both the last PS and middle nodes (middle users and PS) on the random path are equally likely to receive the initiator's requests from any group member. That is, from their perspective, all group members are equally likely to have initiated the URL request (it should be emphasised again that every PS should be separated or they can collaborate to retrieve the initiator's IP address). Although the last PS can verify the group signature with the request to ensure that the initiator has registered, it is difficult for it to reveal the identity of the initiator according to the signature data. Thus, the anonymity of user is strongly protected from PS. It also means that even the PS is compromised by attacker, it does not effect user's anonymity.

iii. Repudiation

As mentioned above, an important advantage of a SAWT system is the ability to trace and reveal hostile users who make malicious accesses through the system. If a user can deny what he did, the system is broken. Further more, if a user or several collaborating users can generate a valid signature, they can cheat PS to submit their malicious request by disguising to be a registered user. To resist such attackers, we adopt strong group signature schemes in the system, such as the one introduced in Section 4.1, whose security is based on the difficulty of calculation of ESKLOGLOG and ESKROOTLOG.

Above, we make a security analysis of a SAWT system and consider several security threats. It should be noted that the system offers no anonymity for users from local network manager, which is generally the network gateway or a firewall system. One possible solution of this problem is to encrypt every message between users and PS. However, this may be unapproved in some local area networks with a secure gateway or firewall, since the encryption may affect the normal filter work of the firewall. This question will be considered in the future work.

5. SIMULATION RESULTS

In this section, we present some simulation results of a SAWT system tested in the LAN of our Lab. The simulation includes client software *Web transactor* and software for PS. Here, we use IE 5.0 browser as the source of requests. The user group of our simulation consists of three users, which are all Pentium MMX 166MHz computers running Web transactor. Only two PS are used in our simulation, both of them are PII 350MHz (we suggest several PS in a SAWT system rather than one and the number of PS depends on the scale of the network). The Web server is a fairly busy PII 350, running Windows NT 4.0 and IIS.

Figure 2 shows the mean latencies in seconds of retrieving web pages of various sizes for various path lengths, which indicate the average durations between the time when Web transactor receives the request from the browser and the time when the web page has been send back to the browser. Here, path length denotes the number of middle nodes (middle users and PS) in a path from initiator to end server.

We list the value of the parameters in our simulation as follows

RSA public key pair (n,e)

$n=9765226370213064031505519333190061377201240486245441720727350557804118341048626671559228 41*3091338268453312787228823305928901203693796209429481993565423187954502288583574456353147 57$

$=3018761797834908213792905878901173942292077842041971030017896314132288724698789826279002605462627247949010182094447332916635197171664984537223819156799442239562323204382458406646 37$, length of n is $|n|$ = 600 bits.

$e = 3$

The private key of GMC is

$d=2012507865223272142528603919267449294861385228027980686678597542754859149799193217519335061737375072855088908906736462531781744184418859241533403242683797653961129048120886996180 27$

Parameters for elliptic curve $E_{(A,B)}(GF(p))$ is

$p =2^{192} - 2^{16} - 1 = 6277101735386680763835789423207666416102355444464034447359$, *length of p is* $|p|$ *= 192 bits*

$A = -3$

$B =52179540753117$

The number of points of the curve above is

$\#E_{(A,B)} =177*35463851612354128609241748154359422549459763644458023953$

$=6277101735386680763835789423321617791254378165069070239681$

The coordinates of the base point G is

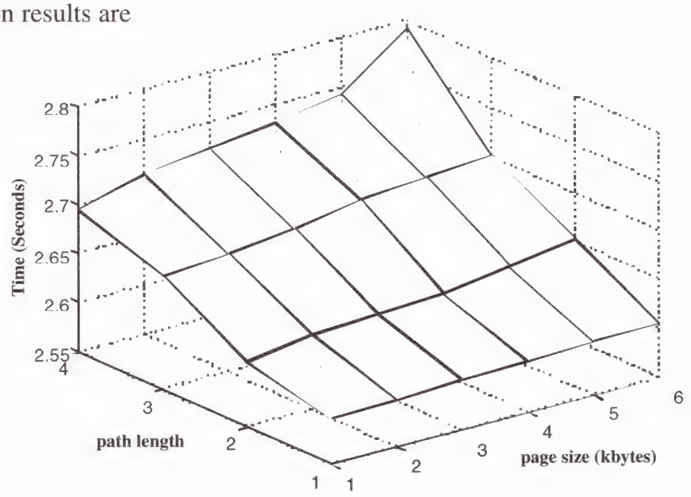
$R_x (G)=5653197914791207271463181835349729564258714540821962188940$

$R_y (G)=3930471266112926226992684258243364565070554531409492318559$

The order of G is $q = 35463851612354128609241748154359422549459763644458023953$

The specific element $a= 2$, which has large multiplicative order modulo both factors of n selected above.

The simulation results are

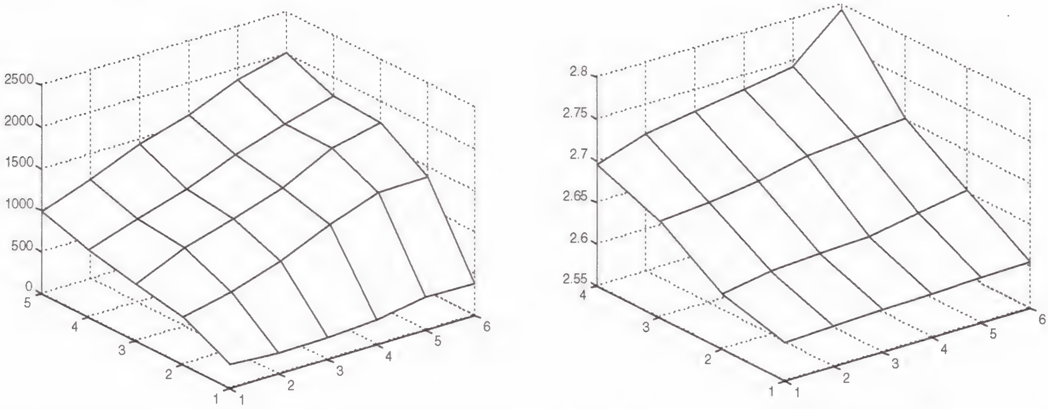


Page size Path length	1k bytes	2k bytes	3k bytes	4k bytes	5k bytes	6k bytes
1	2.597s	2.599s	2.601s	2.602s	2.604s	2.606s
3	2.617s	2.627s	2.631s	2.633s	2.643s	2.654s
5	2.666s	2.671s	2.679s	2.692s	2.696s	2.701s
7	2.695s	2.715s	2.724s	2.733s	2.743s	2.793s

Figure 2: Response latency as a function of path length and page size

It can be easily verified that most latency is due to the group signature. The use of the improved group signature scheme based on ECC has alleviated this problem, but not enough. Therefore, the group signature still remains to be the bottleneck of a SAWT system. In addition, since the real path is established randomly at run time, users cannot choose the path length. In practice, the network is affected by many factors, so the latency is different.

As a reference, we give a performance comparison of SATW system and Crowds system, as shown in Figure 3.



Path length	Page Size (Kbytes)					
	1	2	3	4	5	6
1	2.597s	2.599s	2.601s	2.602s	2.604s	2.606s
3	2.617s	2.627s	2.631s	2.633s	2.643s	2.654s
5	2.666s	2.671s	2.679s	2.692s	2.696s	2.701s
7	2.695s	2.715s	2.724s	2.733s	2.743s	2.793s

Path length	Page Size (Kbytes)					
	0	1	2	3	4	5
1	288	247	264	294	393	386
2	573	700	900	1157	1369	1384
3	692	945	1113	1316	1612	1748
4	814	1004	1191	1421	1623	1774
5	992	1205	1446	1620	1870	2007

Performance of Crowds system

Performance of SAWT system

Figure 3: A comparison of performance of SAWT and Crowds system

We use different definitions of path length and the configuration of the machine (both clients and servers) for the SAWT from the Crowds systems in the test. Thus, the performance charts of these two systems should be presented in two different Figures, as in Figure 3 (More performance details of the Crowds system can be referred elsewhere (Michael, *et al*, 1998)). It is observed from Figure 3 that the mean latency of retrieving a web page is longer in a SAWT system than in a Crowds system. This longer latency of the SAWT system is mainly resulted from the group signature adopted, which has considerable importance in the tracing of malicious connections while ensuring completely anonymity property of the normal surfers. We are now considering the use of a more efficient group signature scheme or other similar approaches to solve this problem.

At the time of writing, the SAWT system is still in testing, with a simulation of the http service calls on port 80. More improvements and simulations of the system will be made in future work.

6. CONCLUSION

This paper proposes a new kind of Secure Anonymous Web Transaction (SAWT) system for anonymous browsing and communication on the World-Wide-Web with high security. In this system, normal users can surf or communicate anonymously while malicious accesses to a Web server will be traced and discovered. The latter property has not been achieved in other existing systems, which can bring more fairness for both users and Web servers. Simulation results and security analysis are presented in this paper, which show the improved security of the system.

Some aspects remain to be developed in SAWT. For example, it is observed that the mean latency of retrieving a web page is longer in our system than in Crowds system. As discussed in Section 5, this problem may be solved by a more efficient and strong group signature scheme or other similar approach.

7. ACKNOWLEDGEMENT

This work is supported by the National Nature Science Foundation of China under the reference number 19931010.

REFERENCES

ANONYMIZER. (2002): The Anonymizer system. http://www.anonymizer.com. Accessed 28-Feb-2002.

CAMENISCH, J and STADLER, M. (1997): Efficient group signature schemes for large groups, *Advances in Cryptology-CRYPTO'97*, 1294:410-424, Springer-Verlag LNCS.

CHAUM, D. (1981): Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84-88.

CHAUM, D. (1983): Blind signatures for untraceable payments. *Advances in Cryptology-CRYPTO '82 Proceedings*, New York and London, 199-203, Plenum Press

CHAUM, D. and E. VAN HEIJST. (1991): Group signatures. *Proceedings of EUROCRYPT'91*, 547:257-265, Springer-Verlag LNCS.

GABBER, E., GIBBONS, P., MARIAS, Y., and MAYER, A. (1997): How to make personalised Web browsing simple, secure, and anonymous. *Proceedings of Financial Cryptography'97*, Anguilla, British West Indies, 1318:17-31, Springer-Verlag LNCS.

JOHNSON, N.F., DURIC, Z. and JAJODIA, S.. (2000): Information hiding: Steganography and watermarking – Attacks and countermeasures, Kluwer Academic Publishers.

KOBLITZ, N. (1987): Elliptic curve cryptosystems, *Mathematics of Computation*, 48:203-209.

KRISTOL, D.M., GABBER, E., GIBBONS, P.B., MATIAS, Y., and MAYER, A. (1999): Consistent, yet anonymous, Web access with LPWA, *Communication of the ACM*, 42(2):42-47.

MILLER, V.S. (1985): Use of elliptic curves in cryptography. *In Advances in Cryptology—CRYPTO'85*, California, USA, 218:417-426, Spring-Verlag LNCS.

REED, M.G., SYVERSON, P. F., and GOLDSCHLAG, D.M. (1998): Anonymous connections and Onion routing, *IEEE Journal on Selected Areas in Communication*, 16(04):482-495.

REITER, M.K., and RUBIN, A.D. (1998): Crowds: Anonymity for web transactions, *ACM Transactions on Information and System Security*, 1(1):66-92.

REITER, M.K., and RUBIN, A.D. (1999): Anonymous web transactions with crowds, *Communication of the ACM*, 42(2):32-38.

BIOGRAPHICAL NOTES

Changjie Wang received the B.S. degree in Telecommunication Engineering in 1996 and the M.S. degree in Communication and Information System in 1999, both from Xidian University, China. He is currently working toward the Ph.D degree in the National Key Lab on. Integrated Services

Networks, Xidian University, China and working as Research Assistant in the Department of Electronic Engineering, City University of Hong Kong.

His general research interests include Internet Security, Cryptography, Secure Mobile Agent System and E-Commerce Security Technology.

Fangguo Zhang received M.S. degree from Applied Mathematics Department, Shanghai Tongji University, China in 1999 and Ph.D degree in Cryptography from Xidian University, China in 2002. He is currently a post-doctor fellow of Cryptology and Information Security Lab, Information and Communications University (ICU), Taejon, KOREA.

His research interests are Elliptic Curve Cryptography, Hyperelliptic Curve Cryptography and Secure Electronic Commerce.

Yumin Wang was born on 18 February, 1936. He received the B.E degree in Dept. of Telecommunication Eng. in Xidian University, China, in 1959. From 1979-1981, he was a visiting scholar in Dept. of E. E. in Hawaii University.

Since 1959, he has been teaching in Xidian University. Currently he is a professor, and a director of the doctoral students in Xidian University. He is a fellow member of the Chinese Institute of Communication, a fellow member of the Chinese Institute of Electronics. He serves as a member of the Board of Governors of the Chinese Institute of Cryptography (preparatory committee) and also serves on the committee of Information Theory Society for the Chinese Institute of Electronics, and a senior member of IEEE.

His research interests are in the general of Communication, Information Theory, Coding and Cryptography.

Information Systems Audit Trails; An Australian Government Survey

Caroline Allinson

Manager Information Security, Information Management Division,
Queensland Police Service, GPO Box 1440, BRISBANE Qld 4001, Australia.
and
Information Security Research Centre (ISRC),
Queensland University of Technology, Brisbane. Queensland. Australia.

Governments have major information holdings on computer systems. This electronically stored information is subject to legislative requirement. However, history has shown that security in relation to the recording of activity against access to information held on Australian government computer systems has been poor and a cause for concern.

A brief definition of information systems audit trails is given, with emphasis on national and international standards requirements. Aspects of Australian privacy legislation are discussed.

Background, detail and results of an Australia wide survey of all government departments is given and contrasted with particular results of a survey conducted by the Australian Commonwealth Privacy Commission four years previous.

It is shown that most organisations studied generate and retain audit trails but the approach is not consistent nor is it comprehensive. Within a four year period there is evidence to suggest that government organisations are increasingly more inclined to generate audit trails. It is also suggested that due to the inadequate and non-compliant security processes and procedures these materials would not withstand a serious legal challenge.

Keywords: Audit-Trails, Evidence, Information-Security, Policy, Survey, Computer.

1. INTRODUCTION

In a computing environment an audit trail supports a management control aimed at enforcing user and system authentication and authorisation. This is achieved by making and keeping, secure records of all necessary information system activities. Two motivating factors for the use of audit trails in the current environment are detection of unethical/unauthorised behaviour and demonstration of a 'proof of business process'. These two factors may include such activities as exceeding access control rights, inappropriate and illicit release of information, correct adherence to required business procedures and fraud prevention (Allinson, 2001). An audit trail record may contain a description of an event/activity, the date and time of the event/activity, the identity of the

Manuscript received: April 2001
Associate Editor: John Roddick

person or sub-system responsible for the event/activity, the location of the individual/system at the time of the event/activity and details of what transpired as a result of the event/activity.

ICL defines audit trails as *"records of those activities considered relevant to the secure and correct running of a system. Audit trailing forms a powerful and important adjunct to authentication and access control. It creates a record of events of the past, ensuring that users can be made accountable for their actions, and more specifically that attempted security violations can be attributed to their source"* Parker and Sundt (1993). The generation of an audit trail showing the timing, sequence and nature of events of interest is essential to systems continuity and ethical behaviour.

Audit Trails have other benefits besides helping to ensure accountability. They also assist in ensuring the integrity of the system itself, i.e. checking that unauthorised changes to software have not occurred, file access controls are properly set and that the communications network has not changed. Checking if the organisation is complying with regulatory controls or to detect suspicious patterns of access such as log-on attempts outside normal hours of business can also be achieved through the generation of audit trails (Caelli, Longley and Shain, 1991).

For both private and public sector organisations the electronic exchange of information via data communication channels has become integral to normal business. The establishment of internal private networks, known as "intranets", within organisations is now standard practice and considered essential to business operation. Extended private networks, known as "extranets" are being established for the sharing of sensitive and non-sensitive information between government organisations. The Internet is rapidly becoming the single most used and essential mechanism for exchange of data and information world-wide.

With the ability for rapid transmission and interchange of not only publicly available information but sensitive and restricted government and business information locally, nationally and internationally there is a greater and more urgent need for information security. There are many definitions for information security and each is dependent upon the context being considered. Amongst other things and for the purposes of this paper, security means authenticity, integrity and privacy. In this context authenticity refers to the verified identification of user to system and of system to user, as well as system to system. It includes the assurance that people are who they say they are, and that messages are attributed to their correct authors. Integrity refers to the assurance that no one has tampered with that message on its way through the network or during storage or passage in a host system. Privacy refers to the assurance to individuals that personal information is protected against improper storage, access, use and disclosure (Caelli, *et al*, 1991; Pfleeger, 1989).

Where systems have been in place for a considerable time it is not uncommon for security controls to be lacking due to the inadequate attention paid to security in previous decades of computer development and use. The Commonwealth Privacy Commissioner in Australia in reference to this rapidly changing technology environment identified that the *"security of personal information within these new environments will need to keep pace with technological change"* Morison (2001).

Audit Trails have been developed for and utilised by Information Systems since the inception of commercial computers. The definition and use of an audit trail has changed over time to reflect the escalation and proliferation of computer systems and networks. One of the most challenging aspects for management is the monitoring, auditing and controlling of activity in a distributed computing environment. There is a requirement for security to be built into all aspects of systems and environments. These information security systems need access and privilege controls, logging and audit controls, accountability controls and monitoring and reporting controls appropriate to the level of sensitivity of the electronically or digitally stored and processed information. Due to the significant increase in computers being used in the commission of crime through their commoditisation, unethical behaviour by authorised users, and business processes requiring electronic contracts or electronic transactions that are legally binding, never before in the history of electronic information processing has the audit trail been as important as it is today. Securing against the loss of audit trail information, and the protecting of audit trails to satisfy legal requirements is a major issue and one that has not been addressed adequately.

Therefore, it is necessary to consider the changing role and importance of audit trails in government organisations. It is suggested that a risk assessment of security threats to government information systems and processes would identify a need for audit trail generation and retention. Computerisation and automation of government business has placed considerable requirement and expectation on proof of process, detection of internal and external attacks such as theft, sabotage or intentional destruction, virus infection and hacking.

Audit trails generated by operational application systems in law enforcement organisations have a four-fold use. Firstly, they assist in assuring that police operations are not compromised and more importantly that the lives of officers working in covert roles are protected from other potentially corrupt officers who have access to the same law enforcement information systems. Secondly, they provide assistance to operational police investigations identifying transactions that were performed in the past that are now important e.g. a vehicle or a person subject of a routine check may now be a suspect in a more serious offence. Thirdly, they are used to assist internal investigations into inappropriate use of information systems. Fourthly, they are used to assist information technology staff in determining reasons for system errors, and restoring data in system recovery processes (Allinson, 2001).

In the state of Queensland audit trails generated from information systems in use by the Queensland Police Service have been presented in legal proceedings as evidence approximately 100 times since 1988. Some of the cases have related to prosecution of Queensland Police officers, others have been presented for verification of a business process i.e. the officer used the system in accordance with standard policy and procedure. Access to these audit trails are restricted due to the sensitive nature of the information stored. All activity is written to the same audit trail, hence, all audit files contain a combination of information that may be unclassified or classified to the highest level. In accordance with the rules of classification the audit trail must be classified and afforded a security level commensurate with the highest sensitivity level of the stored information.

It is expected that other government organisations particularly those operating in the criminal justice arena have similar requirement for audit trails, although perhaps not to the same percentage level of sensitivity. Health service organisations with a need to protect the confidentiality of medical records and computerised systems that administer and/or control medication and assist in life support attract a high degree of risk and consequently a high requirement for security and security monitoring. Organisations dealing with financial and accountancy matters must security monitor for fraudulent activity.

To accommodate this need, security implemented in government organisations must be more than just the basic set of controls relied upon in the past. In the new technology age of open systems inter-connection, security and security monitoring involves the protection of information and data from unauthorised modification destruction or disclosure whether it is intentional or accidental. It also involves protection of programs and operating systems from unauthorised use, modification or destruction, and the protection of information and data transmitted across telecommunications networks.

As previously stated audit trails may also be required as evidence in a court of law and must be able to meet any challenge in relation to the security infrastructure within which they are housed. Therefore they must be generated, retained and presented within an environment where a security framework that is aligned with recognised national and international standards and guidelines has been implemented. This security infrastructure must exist with segregation of audit trail information from the databases and systems for which they are generated. This is especially relevant in cases where intrusion has been detected and the court is being asked to believe an Audit Trail that resides on a system that has been compromised (Allinson, 2001).

Obtaining information on the security of audit trail implementation and assessing such data is of paramount importance and in recent years a number of surveys have been conducted on information technology security and security breaches. International Computers Limited (ICL) and the Gartner group are two of the larger organisations performing this information gathering exercise (NCC, ICL, DTI, 1994; Datapro, 1998). In fact it has become common to conduct surveys on information technology security but very few of these surveys seek to obtain information on audit trails themselves.

Because audit trails are the basis from which most other security initiatives for the monitoring and determination of systems use are derived, it should be mandatory that comprehensive and reliable audit trails be kept for all systems. Most information technology professionals are aware of the need to develop audit trials that cater for the monitoring of system use. However, there appears to be a reluctance to implement systems to cater for this need. As stated in an ICL survey there is a need to "bridge the knowledge gap between awareness of security issues and actually doing something about them" (NCC, *et al*, 1994).

This paper reports the results of an industry survey that was conducted to establish the degree to which Australian government organisations implement security relating to information systems audit trails. It consists of five sections. The first section discusses the results of a survey conducted in 1994 by the Commonwealth Privacy Commissioner and provides background and discussion on audit trails in relation to national and international standards requirements. Section 2 provides an explanation of the industry survey conducted in 1998 including the materials, methods and responses. The third section provides in-depth analysis and reporting of the findings. A comparison of results against two areas of the Privacy Commissioner's survey and the industry survey is reported in Section 4. Section 5 summarises the findings and presents conclusions.

Preliminary findings, which are the subject of this paper, were presented at the "Security In Government Conference 1999" in Canberra Australia and the IFIP WG 9.6/11.7 Working Conference on "Security and Control of IT in Society-II (SCITS-II)" in Bratislava, Republic of Slovakia, 2001.

2. BACKGROUND

Australian Government organisations, the focus of this paper, are governed by legislation, standards and policy in relation to business and service operation. The use of information technology falls within this realm. Legislation are the rules of law within our society that are required to be obeyed. Rules of policy and standards are usually the written control guides. These are considered to be the higher-level instructions indicating intentions about the operations of an organisation. Procedures, sometimes called 'standard operating procedures', are specific operational steps that persons must take to achieve a certain goal. It is important that government organisations give consideration to legislative, standards and policy requirements when implementation of security and security monitoring is being assessed.

Legislation is becoming an important driver within Australia for the implementation of information systems security in both the public and private sectors. Unless an exemption is given in relation to particular sections of legislation it is mandatory for Government organisations to operate in accordance with the law.

One such law, the Privacy Act 1988, is a Commonwealth Act that makes provision for the protection of the privacy of individuals and for related purposes. There are 11 principles set out in Section 14 of the Act. Principles 1, 2 and 3 deal with input controls, i.e. collection and solicitation. Principles 4, 5, 6, 7, 8, and 9 deal with throughput controls, i.e. storage, access, processing. Principles 10 and 11 deal with output or disclosure controls (Australasian Legal Information Institute, 2002).

A breach of security in the use of computer systems presents a much greater threat to privacy than a breach of security in manual systems. Therefore computer professionals, because of the volume of data and the sensitive nature of the data, must be more aware of the level of privacy and the security associated with each level. Principle 4 – Storage and security of personal information part a) states: *"A record-keeper who has possession or control of a record that contains personal information shall ensure:*

a) That the record is protected, by such security safeguards as it is reasonable in the circumstances to take, against loss, against unauthorised access, use, modification or disclosure, and against other misuse."

The words **"against unauthorised access, use, modification or disclosure"** would imply that an audit trail record needs to be kept and that "security safeguards" exist, are clearly documented and understood and may be assessed as being "reasonable" or not. It is the phrase "against unauthorised access" that requires further consideration. A system user who has been authorised to perform particular functions may access and use this information inappropriately. Commissions of Inquiry into inappropriate access and release of government information for gain or corrupt purposes have shown this to be true (CJC, 2000; Fitzgerald, 1989; NSW Ombudsman, 1995). Given this proven fact it is impossible for any government organisation to show that information has not been accessed or used unlawfully or inappropriately by an authorised user unless a secure and reliable audit trail exists.

Principle 6 – Access to records containing personal information, provides for an entitlement by an individual to have access to their personal information that is recorded. An exception to this is where a Commonwealth law may prohibit access. This principle ensures correctness. It also places a greater requirement for understanding and knowledge of the current laws by the computer professional.

A review of previous survey work shows that in 1994 the Australian Commonwealth Privacy Commissioner was concerned about the level of protection of personal information held on government computer systems and the level of compliance with the Privacy Act 1988 by Australian Commonwealth Government Departments. Under his instruction a survey on computer security was conducted which focused on Information Privacy Principle 4. This principle requires agencies to adopt reasonable security safeguards within their information systems. A summary of the comments on the survey findings is listed in Table 1.

One hundred and thirty seven agencies of the 152 receiving the Privacy Commissioner's survey responded. Whilst all areas addressed in the Privacy Commissioner's survey are of interest and relevant to information security the two areas used in this paper for further input and comparison are audit trail controls and computer security policy. The results shown in Table 1 raise considerable concerns in both areas.

Survey Question Area	Comment on Findings
Computer Security Policy	52 % of the population only had a Computer Security Policy (CSP). Overall, the results were disappointing. While the existence of a CSP does not automatically guarantee a secure IT environment, the formulation and maintenance of a policy should focus agencies' attention on the exposure of its systems to potential threat.
Staff Awareness and Training	56% addressed computer security in training sessions. These results were also disappointing. If agencies are to maintain an effective security culture, all systems users need to be aware of computer security issues.
Portable Computing Equipment	A significant amount of personal information is now held on portable computers with a trend toward telecommuting and home based work.
Security Classifications	There is a risk that agencies which overly rely on formal security classifications for protection of personal information may overlook the existence of unclassified material and fail to take adequate protective measures.
Outsourcing	37% employed contractors for the operation of mainframe computer centres or networks. Contractors are generally not covered by the provisions of the Privacy Act. Recommendation has been made that the Act be amended to make a contractor to a federal agency primarily liable for compliance with the Information Privacy Principles as if the contractor were an agency.
Defence Signals Directorate liaison	The survey results clearly show a low level of consultation with DSD and a lack of awareness of DSD's role. Also, few agencies have adopted the standards embodied in DSD instructions, relating to authentication identifiers and communications security.
Networks and Communication Links	80% use Local Area Networks and 55% use wide area networks. However, only 12% use encryption on nationally-linked communications paths and 4% on regional, State and central office facilities. The low level of such protection across all agencies is an issue which must be addressed.
Physical Security	The results for this section appeared reasonably satisfactory although the survey was not designed to assess the adequacy of these measures or their quality.
Audit Trail Controls	Results confirm that audit trail controls are more prevalent in mainframe based systems than in networks and networked desk-top environments. Given the finding that 80% of agencies use Local Area Networks and 55% use Wide-area networks the relative lack of audit trail controls on networks is of some concern.
Audit Programs	Results indicate that over half of the survey population had not conducted internal computer security or access audits since 1 July 1992 and 47% of agencies have not included such audits in their audit plan for 1993/1995. This is of considerable concern.

Table 1 Summary of Findings for the Privacy Commissioner's Survey (Morison, 2001)

2.1 National and International Standards

In the early to mid 1980's The United States Department of Defence published the *Trusted Computer System Evaluation Criteria* (TCSEC) series of documents against which computer systems could be evaluated for security. TCSEC, although now replaced by the Common Criteria (CC), International Standard IS 15408, as discussed later is a standard that has been recognised and used internationally for evaluating the effectiveness of security controls built into computer systems. The Criteria are divided into four divisions D, C, B, and A. These divisions are ordered hierarchically with the highest level of security assurance being A. In divisions C and A there are classes denoted by numerical sequence, i.e. C1, C2, B1, B2 (NCSC, 1985).

For systems that meet criteria C2 through A1 there is a requirement that a user's action be open to scrutiny by means of an audit mechanism. *A guide to Understanding Audit in Trusted Systems* was also published by the United States government to assist manufacturers on how to design and incorporate an effective audit mechanism into their systems and to assist information systems implementers in the effective use of systems where audit trails are provided as part of the software package (NCSC, 1987).

TCSEC has an Accountability Control Objective which states: *"Systems that are used to process or handle classified or other sensitive information must assure individual accountability whenever either a mandatory or discretionary security policy is invoked. Furthermore, to assure accountability the capability must exist for an authorised and competent agent to access and evaluate accountability information by a secure means, within a reasonable amount of time and without undue difficulty".*

The associated control objective for auditing states: *"A trusted computer system must provide authorised personnel with the ability to audit any action that can potentially cause access to, generation of, or effect the release of classified or sensitive information. The audit data will be selectively acquired based on the auditing needs of a particular installation and/or application. However, there must be sufficient granularity in the audit data to support tracing the auditable events to a specific individual (or process) who has taken the actions or on whose behalf the actions were taken"* NCSC, (1987).

Like most areas pertaining to IT, the development of security evaluation criteria for international compliance and use had little uniformity. The international information technology standards bodies recognised the need for a uniform scheme and collaborated to develop what has become known as the 'Common Criteria' resulting in international standard ISO/IEC 15408-1 Information Technology – Security Techniques – Evaluation Criteria for IT Security (International Standards Association, 1999). It is stated in the Common Criteria (CC) that *"Security, auditing involves recognising, recording, storing, and analysing information related to security relevant activities. The resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them"* (International Standards Association, 1999). The CC can be used to select the appropriate IT security measures and it contains criteria for evaluation of security requirements. Its main target audience is consumers, developers, evaluators, and others such as system custodians, security officers, auditors, security architects and designers.

The CC has a uniform construct that is organised into security requirements in a heirarchy of class, family and component. The functional requirements for the security audit class, within the CC, consists of five components. These components provide for security audit automatic response, security audit data generation, security audit analysis, security audit review, security audit event selection and security audit event storage.

The Australian and New Zealand Standard for Information Security Management (Standards Australia, 2001) Section 9.7 provides guidance for monitoring system access and use. Whilst not as comprehensive as it could be it does provide for event logging and procedures for monitoring use of information processing facilities. It does not provide for the handling of audit trails and the security mechanisms required to protect and secure them.

2.2 Classification of Information

As shown in the national and international standards, there is a requirement for auditing of sensitive information. The definition of "sensitive" is defined for Australian government agencies in policy that relates to the classification of information. Both commonwealth and state governments in Australia have either developed or are in the process of developing policy for information classification. However, it is doubtful that full implementation of these policies has been effected. This would indicate that many government organisations do not know the extent of their sensitive and classified information holdings. The Commonwealth Protective Security Manual (PSM) was developed to reflect the minimum standards for security management within Commonwealth Government Departments (Commonwealth Government, 1998). The Australian Communications-Electronic Security Instructions 33(ACSI 33) provides detailed instruction in relation to the handling of audit trails (DSD, 1998).

As guidelines for police agencies a 'Standard Law Enforcement Information Security System' manual was developed in 1994 by the then National Police Research Unit (NPRU) now known as the Australasian Centre for Policing Research (ACPR). This manual has a section, adapted from the PSM on information classification. To date few, if any, state police agencies have implemented this classification system to its intended full use. Most information systems have not been developed to include classification of data and information stored electronically (ACPR, 1994).

The model used within these two manuals provides a distinction between information that relates to national security and information that is considered as other sensitive material.

National Security	Sensitive Material
TOP SECRET	
SECRET	HIGHLY PROTECTED
CONFIDENTIAL	PROTECTED
RESTRICTED	IN-CONFIDENCE

Table 2: Classification Comparison

Sensitive government information must be classified as either HIGHLY PROTECTED, PROTECTED or IN-CONFIDENCE. The lowest level of classification for sensitive information is IN-CONFIDENCE. The IN-CONFIDENCE security classification must be assigned for:

a) information concerning the private affairs of individuals (eg staff records, medical records, customer/client records);
b) information provided to agencies under an assurance/expectation of confidentiality;
c) tender documents;
d) sensitive industrial relations matters; and
e) compilations of information which individually are unclassified but which collectively should be classified In-Confidence.

If consideration is given to this classification requirement and the Privacy Legislative requirement, and then compared to the requirements for audit trails in the security evaluation criteria for sensitive information, it is obvious that Australian government organisations require audit trails for sensitive information at all levels including IN-CONFIDENCE level.

3. SURVEY OF AUSTRALIAN GOVERNMENT ORGANISATIONS

To obtain a better understanding of the current security and audit practices of all Australian government organisations, a survey which focused on information systems audit trails was developed. The survey was interested only in audit trails for "traditional" information systems involving computers and data networks.

The survey instrument was a mailed, printed questionnaire which was sent to Australian Commonwealth and State government organisations. In the State of Queensland related departments were included, i.e. "quasi government" organisations.

The aim was to deliver information on organisational process and procedure for the generation, retention, storage, use and control of information systems audit trails.

All Australian Government organisations were chosen and were drawn from the Brisbane State Library reference list. Addresses were verified via telephone directory listings. Not all related Departments were included.

3.1 Materials, Method and Response

Three hundred and ninety organisations were identified as the population for survey purposes. Twenty surveys were sent to Commonwealth Departments and related Departments and 370 were sent to Queensland State Government Departments and other related State Government and Territory Departments throughout Australia, (refer to Figure 1). The number of responses became the sample size.

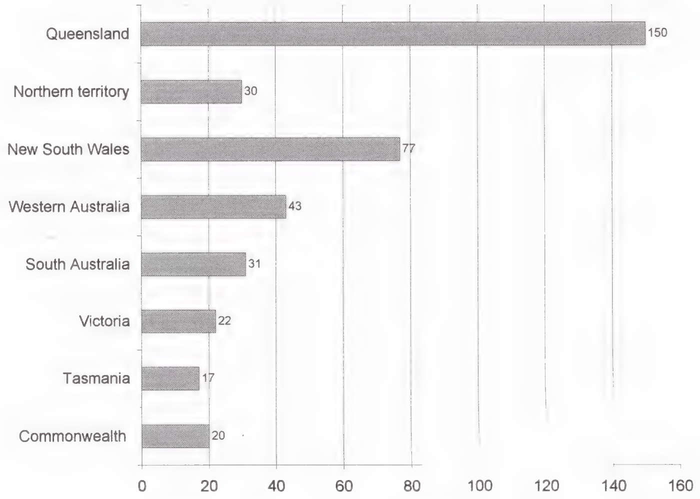


Figure 1: Survey Recipients

One hundred and eighteen responses were received but one was blank and although counted in the initial entry it was discarded in the calculations. Hence a sample size of one hundred and seventeen was determined by the responses (Figure 2). A 30% sample size was considered more than adequate as most statistical analysis is performed on a sample size of 10% to 20% (Harrison and Tamaschke, 1984).

Categorisation of each organisation was performed by firstly dividing responses into Commonwealth and State/Territory groups and then further dividing them to determine Law Enforcement, Law Enforcement affiliated and other participants in the criminal justice system. Of the 117 respondents 20 were Commonwealth, 97 State, and of the 20 Commonwealth responses 3 were from Law Enforcement affiliate. Of the 97 State responses 3 responses were from Law Enforcement, 3 from Law Enforcement affiliated and 7 from criminal justice system members. Hence 16 organisations were involved in the criminal justice arena.

Organisational size was determined by the number of people authorised to use the information systems. This was given the term "user base". As depicted in Figure 3, good cross representation of organisational size was received with approximately one third representation from small, medium and large organisations. "Small" was considered less than 100 users, "medium" was considered between 100 and 1,000 users and "large" was considered to be an organisation with any number of users greater than 1,000. Cross analysis showed that 50% of the responses from Commonwealth Departments were in the large organisation category.

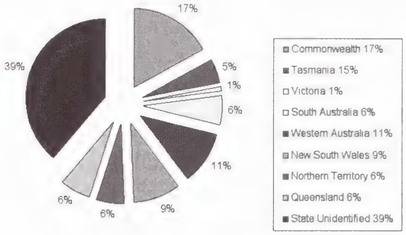


Figure 2: Survey Respondents

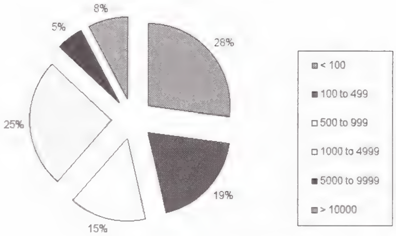


Figure 3: User Base

The SAS Institute Incorporated statistical package SAS version 6.12 running on an HP-UX operating system was used for data collation and analysis. It was not intended that complex statistical methods would be used. Frequency analysis, overall percentage calculation, and cross tabulation where testing proportions for two populations was required, were used. For the comparison of findings and hypotheses testing with the results of the Privacy Commissioner's survey, a two tailed chi-square goodness of fit test and a one sample test on the proportion using normal distribution was used.

The survey was divided into 5 sections:

Section One addressed Audit Trail generation and retention. This section sought to deliver information on:

- retention and retention period,
- the existence of a single audit facility,
- how consistently audit trails were implemented,
- the way in which audit trails are written or generated,
- reliance on application driven audit trails,
- generation and reliance on audit trails for office systems and E-Mail,
- the ability to inhibit audit trails and to what extent system use is recorded.

Information was also sought on what legislative requirement is the driver for audit trail generation and retention.

Section two sought to deliver information on storage and backup of audit trails. This included:

- the storage medium used,
- the employment of enhanced security mechanisms such as encryption
- the employment of enhanced security mechanisms such as checksum verification
- backup process and procedure
- the type and number of backups.

Section three addressed the purpose and use of audit trails to include the reconstruction of activity performed, if audit trails had been produced as evidence in a court of law and the purpose for which they are used.

Section four addressed responsibility and control for audit trails. This section sought to deliver information on:

- the personnel role and function as well as the department responsible for audit trails,
- if responsibility is included in job descriptions,
- the generation of audit trails to monitor technical support activity
- inclusion of policy and procedure for audit trails in security policy.

Section five sought to obtain basic information about the organisation including the type and size of the organisation.

3.2 Results and Analysis

Of the 117 responses received 100 said they generate audit trails. Of these 100, 96 said they retain them (Figure 4). Thirty-five percent retain indefinitely, with a further 34% retaining audit trails for between 5 to 10 years. Law Enforcement and criminal justice system organisations retained for 5 years or greater. There were fewer organisations retaining for a shorter period of time although of the 24 organisations retaining for less than 5 years, 6 said they retained audit trails for 1 year or less. Six organisations responded to the option of "other" and specified that audit trail procedures varied depending on outsourcing arrangements and the systems involved (Figure 5).

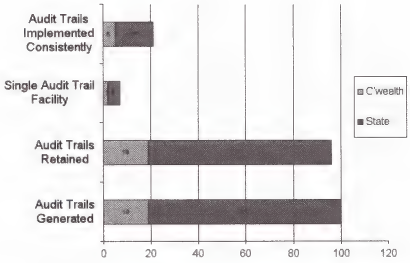


Figure 4: Audit Trail Generation & Retention

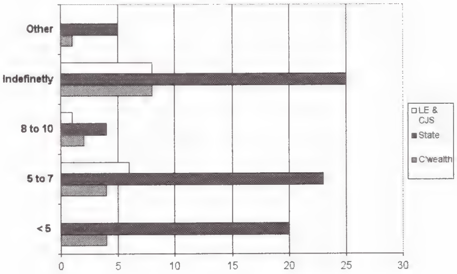


Figure 5: Audit Trail Retention Period

It is reasonable to say that from these results, government organisations in Australia have a need and/or are required to generate and retain audit trails for a significant period of time. Concern is raised for those 17 organisations not generating and those organisations holding audit trail information for short periods of time only.

Although required to generate and retain audit trails the manner in which this is done is inconsistent. Eighty-six organisations said they do not have a single audit trail facility implemented and 64 of these do not implement their audit trails consistently (Figure 4).

There is a strong reliance on in-house developed, application driven audit trails. This question attracted a 73% response. A pertinent survey question, question six may have confused respondents a little by listing "operating system" journals and "system" journals separately. In most situations these two may be the same. Given this, the combination of these show an 85% reliance. This is a strong reliance on operating system journals. There is a 49% reliance on database journals with 25% only reliance on audit trails from network/communication nodes (Figure 6).

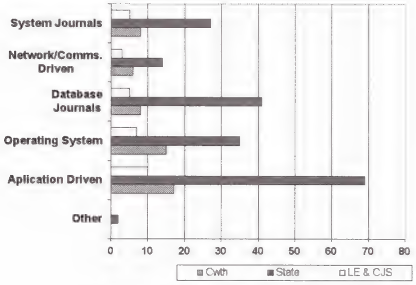


Figure 6: Audit Trail Reliance

Of the 87 organisations relying on application driven audit trails, 61 developed them in-house with 52 purchasing proprietary software. Twenty-six organisations rely on a combination of both.

Where used, Microsoft Corporation's 'Windows NT' is relied on for audit trail generation for office systems with 15% auditing word processing activity and 14% auditing spreadsheet activity. A larger number of organisations than expected are generating Audit Trails for office systems. The number generating audit trails for E-Mail was also greater than expected with 34% saying they audit internal E-mail and 37% "Internet" E-mail. This is interesting given most E-Mail systems do not have full auditing capability built in (Figure 7).

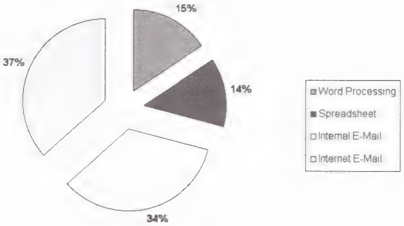


Figure 7: Office Systems and Email

In Figure 8 it can be seen that the ability to inhibit audit trails was high with only 29% stating they had no ability to inhibit. Most inhibiting is done at system level. Data, transaction and user level inhibiting were not favoured, with 8% only in each category claiming this could be done. The ability to inhibit audit trails at any level raises concerns over the ability to prove if an audit trail system was operating at a given time.

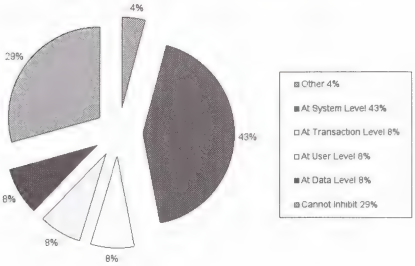


Figure 8: Ability to Inhibit Auditing

Recording of system use and activity appears to be inconclusive as only 13 respondents claim to record every keystroke; 31 record every query but in section three, 64 said they could reconstruct all activity performed on a data element (Figure 9). It is not possible to reconstruct all activity unless queries have been recorded. Cross tabulation showed that of the 13 who record every keystroke, 8 record every addition, 8 record every deletion, 9 record every modification and 7 record every query. At best 7 organisations only could reconstruct all activity. A check against 55 organisations who said they record activity on particular systems only, showed that 44 record every addition and deletion 48 record every modification and 21 record every query. At best only 21 organisations can reconstruct all activity.

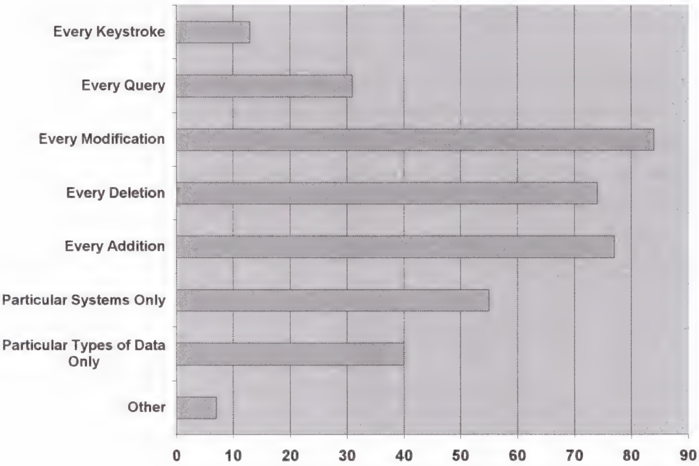


Figure 9: Recording of Activity and System Use

Thirty-one organisations, made up of 6 from the Commonwealth and 25 from the States, said they have legislative reason for creating and retaining audit trails. It is interesting to note that Legislation applicable to Finance and Audit is still the main reason for maintenance of audit trails, with 25 of the 31 giving this reason. Several of the respondents cited more than one Act of State or Federal government. One only organisation said they keep audit trails for reasons of appropriate and relevant privacy legislation (Figure 10). Replies to the questionnaire in this area beg the question as to whether respondents were aware of their legislative requirements. All organisations working under privacy legislation should have said that they are subject to the legislation, and should maintain audit trail data. However, do they really know? This illustrates the possibility of a knowledge gap.

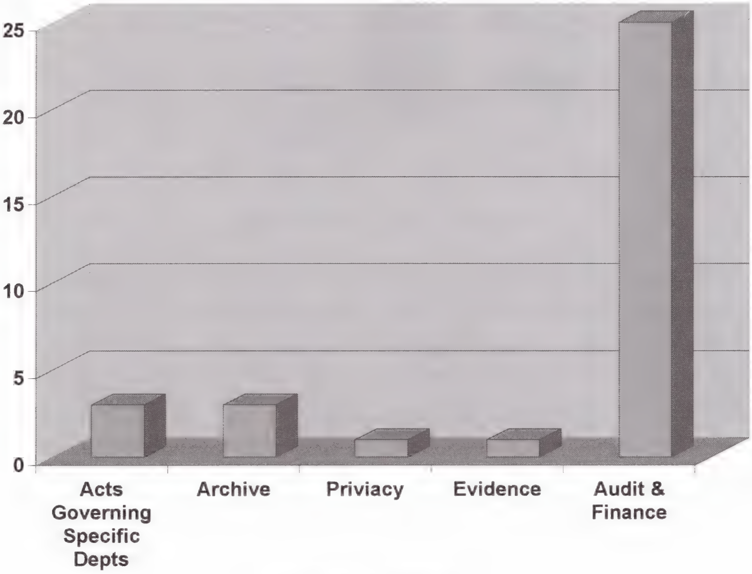


Figure 10: Legislation

Magnetic tape is the main medium used for the storage of audit trails with 36% preferring this form. Twenty percent stated more than one type of storage medium in use. Two organisations store audit trails in hard copy form and 4 used CD ROM (Figure 11).

There is little segregation of audit trails, and enhanced security mechanisms such as check summing and encryption are rarely used (Figure 12). Backup activity is taken a little more seriously with 50% implementing 2 copies with one of these maintained off site. Thirty-one percent of these organisations secure 1 copy as the original.

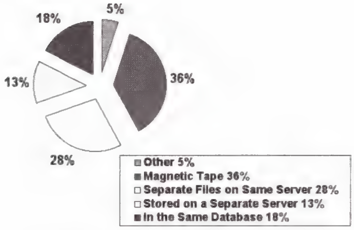


Figure 11: Storage of Audit Trails

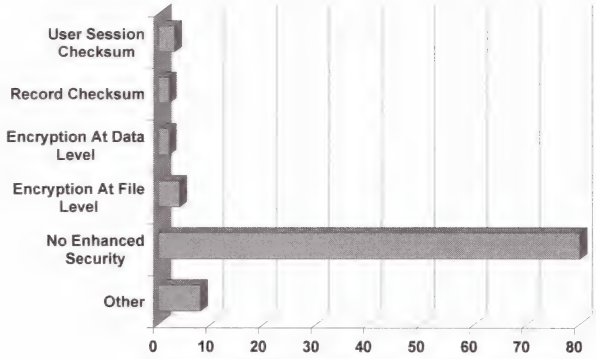


Figure 12: Enhanced Security Mechanisms

As previously shown there is an inconsistency in the recording of activity and the belief that all activity can be reconstructed. Fifty-five respondents said they could reconstruct all activity for a user, 22 could reconstruct all activity performed from a given terminal, 25 from a particular location and 64 claimed that they can reconstruct all activity on a data element. Given the inconsistency, the number of organisations claiming they can reconstruct all user activity may be less than shown in Figure 13 but the results do indicate a recognition for the need to be able to identify not only who performed the activity but from what location and on what data.

Twenty-eight organisations claim to have been required to produce their audit trails as evidence in court. Sixteen of these organisations indicated they were Law Enforcement groups or Law Enforcement affiliated or part of the criminal justice system. This would indicate that there is a need for other organisations to have security controls consistent with those in Law Enforcement for evidentiary purposes (Figure 14).

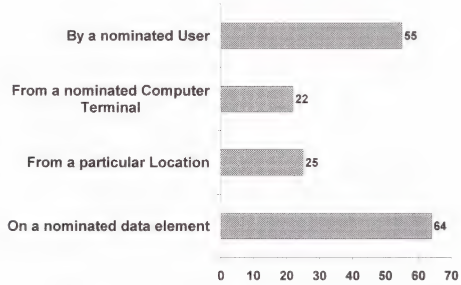


Figure 13: Reconstruction of Activity

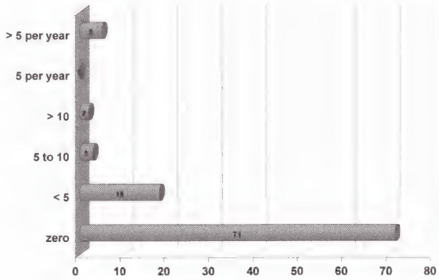


Figure 14: Audit Trails as Evidence

Whilst acknowledgement of the requirement to produce audit trails in court was made by 24%, which is “moderate” to “low”, the main reason given for keeping audit trails is technical. Sixty-three percent of organisations said that they keep audit trails for such technical reasons. Fifty percent of organisations said they keep audit trails for business/operational purposes, 58% for inappropriate use of systems, 48% for pro-active monitoring and 15% for freedom of information requests (Figure 15). It is predictable that the Internet driven electronic business and commerce direction of information technology will cause this requirement to increase and the need to produce audit trails in court to also increase. There is concern for the low number of organisations not pro-actively monitoring system use.

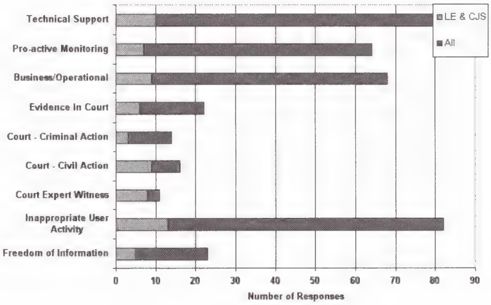


Figure 15: Purpose and Use of Audit Trails

The results in Figure 16 show that technical information systems staff have the greatest responsibility for audit trails. This accounts for 38% of respondent organisations. More than half have no responsibilities written in job descriptions; 45% have no clear lines of segregation; 47% do not monitor technical support staff; and 33% only have a policy for Audit Trails (Figure 17). The responsibility and control for audit trails is poor from a security perspective.

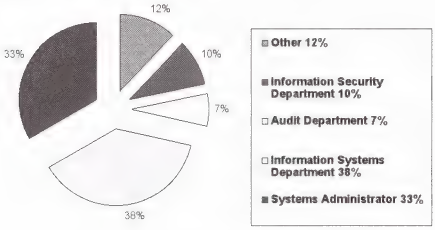


Figure 16: Responsibility for Audit Trails

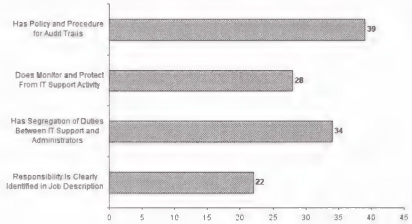


Figure 17: Audit Trail Controls

4. SURVEY COMPARISON

Comparison between the two surveys, that being the privacy survey conducted on information security by the Privacy Commissioner in 1994 and the whole of government survey conducted on audit trails in 1998 was performed in two areas. The first comparison sought to determine if government organisations were including policy for audit trails in their information security policies. The second sought to determine if organisations, between 1994 and 1998, had improved their willingness to generate audit trails. Two hypotheses were developed, one to test the distribution of the variables in the two populations and another to test the value of a population proportion.

	Privacy Survey	**Industry Survey**
Population =	152	390
Sample =	137	117

Hypothesis 1: Where information security policy exists, audit trail policy is not always included.

Fifty-two percent of organisations responding to the Privacy Survey stated they have an information security policy. Thirty-three percent of the organisations responding to the industry survey stated they have and audit policy.

There was a difference in the proportion of those who had audit policy compared to those with information security policy.

$$(\chi^2 = 16.5, \text{p-value} , 0.0001, \text{df}=1)$$

This difference is highly significant showing that organisations are significantly more likely to have an information security policy than an audit policy. Hence, acceptance of Hypothesis 1 is possible.

Hypothesis 2: The percentage of government organisations who are inclined to generate audit trails for their information systems has increased (1994 – 1998).

Sixty-one percent of organisations responding to the Privacy Survey stated they generate audit trails. Eighty-two percent of the organisations responding to the industry survey stated they generate audit trails.

There was a difference in the proportion of those who responded to the whole of government survey and those who responded to the privacy survey. The difference was highly significant. Those responding to the whole of government survey were significantly more likely to generate audit trails.

$H_0 : \pi = .61$
$H_1 : \pi > .61$
($z = 4.67$, p-value < 0.001)
Reject H_0, Accept H_1

The proportion generating audit trails (in 1998) has increased from that established in 1994, hence there is evidence to suggest that Hypothesis 2 is correct and therefore can be accepted.

5. SUMMARY

Broadly the survey has shown deficiencies in all areas surveyed and these deficiencies are serious. Although it has been shown that most organisations that were studied, generate and retain audit trails the current practices employed by Australian Government Organisations are neither consistent nor comprehensive.

Concern is raised in relation to those organisations not generating audit trails and for those generating but retaining for short periods of time. Non-compliance with privacy legislation is evident with suggestion that many organisations are not aware of their legislative requirement.

From the results shown it is reasonable to say that government organisations in Australia have a need for or are required to generate and retain audit trails for information systems. It is also predictable that the increased use of electronic business and commerce at government level will cause this requirement to increase.

It has been shown that organisations other than Law Enforcement are required to produce audit material as evidence, given the lack of security practice and administrative control it is suggested that if tested in a court of law the audit trails would not withstand a serious legal challenge.

ACKNOWLEDGEMENTS

This paper has been edited and reviewed by Professor William Caelli, Head of School of Data Communications, Queensland University of Technology (QUT), Brisbane, Australia.

REFERENCES

ALLINSON, C. L. (2001): Information systems audit trails in legal proceedings as evidence, *Computers & Security*, Vol 20 Number, Elsevier Advanced Technology,UK.

AUSTRALASIAN LEGAL INFORMATION INSTITUTE, (2001): Privacy Act 1988 – Notes, http//www2.austlii.edu.au/privacy/Privacy_Act_1988/index-Division-2.html. Accessed 2001.

CAELLI, W., LONGLEY, D. and SHAIN, M. (1991): Information Security Handbook, Macmillan Publishers Ltd, Great Britain.

CJC, (2000): Protecting confidential information, *Criminal Justice Commission*, Brisbane.

COMMONWEALTH GOVERNMENT OF AUSTRALIA, (1998): *Protective Security Manual*. Commonwealth Government of Australia. Edition 3, 1998.

DATAPRO, (1998): 1998 Worldwide survey on information security issues, GartnerGroup, USA.

DEFENCE SIGNALS DIRECTORATE, (1998): Australian communications-electronic security instructions 33(ACSI 33); Security Guidelines for Australian Government IT Systems.

FITZGERALD, G. E. (1989): Report of a commission of inquiry pursuant to orders in council, The Government Printer Queensland, Australia.

HARRISON, S. R., and H. U. TAMASCHKE, H. U. (1984): Applied statistical analysis, Prentice-Hall of Australia Pty Ltd, Australia.

INTERNATIONAL STANDARDS ORGANISATION, (1999): Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model, Reference number ISO/IEC 15408-1:1999(E), First Edition.

MORISON, J. (2001): Computer Security – a survey of 137 Australian agencies, privacy law & policy reporter, Volume 3, Number 4, http://www.austlii.edu.au/au/other/plpr/vol3/vol3No04/v03n04c.html. Accessed 2001.

NCSC, (1985): DoD trusted computer system evaluation criteria, Department of Defence (DoD), DoD 5200.28-STD.

NCSC, (1987): A guide to understanding audit in trusted systems, National Computer Security Center, Maryland, USA.

NCC, ICL, DTI, (1994): The IT security breaches survey, The National Computing Centre Limited.

NCPR, (1994): A standard law enforcement information security system, National Police Research Unit.

NSW Ombudsman, (1995): Confidential information and police, Office of the Ombudsman, NSW.

PARKER, T., and SUNDT, C. (1993): Information security handbook ICL, International Computers Limited, England.

PFLEEGER, C. P. (1989): Security in computing, Prentice-Hall Inc., United States of America.

STANDARDS AUSTRALIA, (2001): AS/NZS ISO/IEC 17799:2001 Information technology – Code of practice for information security management.

BIOGRAPHICAL NOTES

Caroline Allinson is employed as Manager Information Security for the Queensland Police Service in Brisbane, Australia.

This position involves management of information security policy development, information systems access control, assisting with investigations which include evidence in court, security auditing and security advice and consultancy.

In September 1994, Ms Allinson was awarded the Courier-Mail Police scholarship and travelled internationally to research and study Information Security and Computer Crime.

Ms Allinson is a Certified Information Systems Auditor, is a Bachelor of Business (computing), a Master of Information Technology, and is currently studying for her Doctorate. Her thesis is titled "Legislative and Security Requirements of Audit Material for Evidentiary Purpose".

She is a member of the Australian Institute of Management, the Australian Computer Society, the Security Industry Professionals Technical Association (ISSA) and The Information Systems Audit and Control Association .

Ms Allinson has worked as a guest lecturer and tutor in Information Technology at the Queensland University of Technology, Brisbane, Australia and is a member of the Faculty of Information Technology's Advisory Committee.

Submission Guidelines

The Journal encourages submission of innovative and original articles in all areas of Information Technology including Computer Science, Software Engineering, Information Systems, Computer Systems and Information Engineering and Telecommunications.

The following forms of article are published:

- Full articles examining original research and/or application;
- Short articles describing original research and/or application or commenting on relevant legal, political or technical innovations;
- Survey or tutorial articles describing new directions in Information Technology or which are relevant to the practical application of fundamental research;
- Book Reviews;
- Announcements; and
- News Briefs.

Full details can be found at the Journal home page: www.jrpit.acs.org.au

Subscription Guidelines

The annual subscription for the Journal is

Individuals	
Australian Computer Society members	Free
Delivery address in Australia, incl. postage and GST when applicable	A$66
Overseas, incl. postage.	US$54
Institutions	
Australia, including postage and GST where and when applicable.	A$99
Overseas, including postage	US$81

All subscriptions to the Journal are payable in advance and should be sent (in Australian Currency) to the address below.

Bibliographic Information

Journal of Research and Practice in Information Technology – *J.Res.Prac.Inf.Tech.*
formerly the Australian Computer Journal (ISSN 004-8917)

ISSN	1443-458X
Circulation	15,500
Publisher	Australian Computer Society Inc. PO Box Q534, QVB Post Office Sydney 1230, New South Wales AUSTRALIA
Production Management	Associated Business Publications Pty. Ltd. P.O. Box 3267, Umina Beach, NSW 2257, Australia
Printer	Mailing and Print Services Pty Ltd 8 Aquatic Place, Frenchs Forest, NSW 2086, Australia
Cover Design	Modern Planet Design (08) 8340 1361 • modern@webmedia.com.au

Contents

Special Features

A publication of the Australian Computer Society
ISSN 1443- 458X